(1)

National Defence / Défense nationale

AD-A276 218

DTIC
ELECTE
FEB 28 1994
C

EVALUATION OF HardSys/HardDraw, AN EXPERT SYSTEM FOR ELECTROMAGNETIC INTERACTIONS MODELLING

by

Marc Dion

DEFENCE RESEARCH ESTABLISHMENT OTTAWA

REPORT NO. 1175

Canada

May 1993
Ottawa

National Defence | Défense nationale

Accesion For	
NTIS CRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

EVALUATION OF HardSys/HardDraw, AN EXPERT SYSTEM FOR ELECTROMAGNETIC INTERACTIONS MODELLING

by

Marc Dion
Nuclear Effects Section
Electronics Division

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
REPORT NO. 1175

PCN
041LT

May 1993
Ottawa

ABSTRACT

The Department of National Defence and the National Research Council have sponsored the development of HardSys/HardDraw, an expert system for the modelling of electromagnetic interactions in complex systems. This report gives a description of HardSys/HardDraw and reviews the main concepts used in its design. Various aspects of its implementation, user interaction and modelling concepts are evaluated. Some deficiencies are identified and enhancements are proposed to overcome them. Concepts of uncertainty are reviewed and an approach using confidence factors and fuzzy arithmetic is developed. A new method relating both the frequency and time domains is presented and is applied for the calculation of failure index and shielding effectiveness.

RÉSUMÉ

Le département de la défense nationale et le centre national de recherche ont développé un système expert pour prédire les interactions électromagnétiques dans des systèmes complexes. Ce rapport donne une description de ce système et des concepts utilisés. Plusieurs aspect de son implémentation, de son interaction avec l'utilisateur et des modèles utilisés sont discutés. Certaines déficiences sont identifiées et plusieurs améliorations sont proposées. Le concept d'incertitude est présenté et une approche utilisant les facteurs de confiance et l'arithmétique floue est présentée. Une approche innovatrice pour relier les domaines fréquentiel et temporel est présentée et est appliquée pour le calcul des indices de défaillance ou des coefficients de blindage électromagnétique.

EXECUTIVE SUMMARY

The Department of National Defence and the National Research Council have sponsored the development of HardSys/HardDraw, an expert system for the modelling of electromagnetic interactions in complex systems. It consists of two main components: HardSys and HardDraw. HardSys is the advisor part of the expert system. It is knowledge-based, that is it contains a database of models and properties for various types of electromagnetic interactions. Problems are defined by using the electromagnetic topology concept. HardSys takes into account the characteristics of electromagnetic emissions, the shielding effectiveness and the susceptibility of components to calculate the likelihood of failure of the system. HardDraw is a powerful drawing tool used to create or modify the electromagnetic topology of a problem and to create automatically the associated flow graph.

This report gives an evaluation of HardSys/HardDraw. It analyses its implementation, user interaction and modelling concepts. Some deficiencies are identified and enhancements are proposed to overcome them. Some aspects of the representation of the electromagnetic quantities are found to be inadequate and an alternate representation is proposed.

Concepts of uncertainty and fuzzy arithmetic are reviewed and an approach based on confidence factors and fuzzy arithmetic is developed.

A new method for relating the frequency domain with significant time domain characteristics, such as peak value, rise time and duration, is presented. This method is applied for the calculation of failure index and shielding effectiveness. It is also suitable for the calculation of the optimal additional shielding required to protect components against upset or damage, based on threshold criteria defined in terms of voltage, current, power, energy or duration.

Preceding Page Blank

TABLE OF CONTENTS

Preceding Page Blank

1.0 INTRODUCTION

In the design process of electronic systems, it is necessary to assess the susceptibility (or vulnerability) of the electronics to natural and man-made electromagnetic sources such as lightning, nuclear electromagnetic pulses (NEMP) and high-power microwave from radars and directed-energy weapons; and to provide added protection to the sensitive components if the system is to survive under a given environment. To support this hardening process, the National Research Council (NRC) and the Department of National Defence have sponsored the development of HardSys/HardDraw, a tool for the modelling of electromagnetic interactions in a system. HardSys/HardDraw has been designed to use expert system techniques to help the design engineer to adequately harden electronic systems against electromagnetic threats.

This report first gives a description of HardSys/HardDraw and reviews the main concepts used in its design. It then summarizes our evaluation of the system. Various aspects of its implementation, user interaction and modelling concepts are discussed. Finally, enhancements are proposed to address deficiencies and shortcomings of the algorithms or models used by the system. Most of these enhancements are already at some stage of development.

1.1 DESCRIPTION

A comprehensive description of HardSys/HardDraw is found in the final report [1] and a good summary is found in [2]. It is not the intent to repeat herein this description, but a brief summary will be given, particularly on aspects which will be further developed in this report.

HardSys/HardDraw is an expert system for the modelling of electromagnetic interactions in a system. It consists (from the user point of view) of two main components: HardSys and HardDraw. HardSys is the advisor part of the expert system. It is knowledge-based, that is it contains a database of models and properties for various types of electromagnetic interactions. Problems are defined by using the electromagnetic topology concept (Section 1.1.1). HardSys takes into account the characteristics of electromagnetic emissions, the shielding effectiveness and the susceptibility of components to calculate the likelihood of failure of the system. HardDraw is a powerful drawing tool used to create or modify the electromagnetic topology of a problem and to create automatically the associated flow graph. HardDraw also implements all the functions related to user interaction.

1.1.1 Electromagnetic Topology

The problem of predicting the response of a complicated system to incident electromagnetic fields has been the subject of considerable study, but it is still impossible (except for the simplest cases) to perform a rigorous analysis of the system. It is therefore desirable to divide a problem into smaller subproblems (*divide and conquer* approach) to obtain a reasonable estimate of the solution. To solve this problem, the concept of electromagnetic topology was developed and studied by a number of authors [7]-[11]. With this concept, a system is decomposed into a set of volumes or surfaces which are interconnected to describe the propagation of the electromagnetic energy. The resulting topological diagram can be solved with graph theory to obtain a solution for every node. The surface-based approach described in [9] models a system in terms of surfaces (nodes), such as the inside and the outside of a body, which are separated by volumes, in this case, the air or the metal within a physical shield. The nodes are interconnected by branches representing the propagation between the two surfaces on either side of a volume (diffusion through a shield, propagation through apertures, etc.) or between two surfaces separated by air (electromagnetic radiation, cable connections, etc.). The volume-based approach used by HardSys treats the volumes as the primary objects (nodes), interconnected with branches representing the transfer function from one volume to another. Each branch has an additional node representing the surface which it penetrates (forming a bipartite graph as described in [1]). Both approaches can adequately model electromagnetic problems, although the volume-based definition is more intuitive and suitable to implement into a topology drawing tool such as HardDraw, while the surface-based definition gives a little more flexibility when specifying connectivity. For instance, with the volume-based model, the propagation of an ambient field in volume V_0 into a shielded enclosure V_1, is represented by a single branch which has to take into account the propagation through the air medium as well as the penetration characteristics of the shield. By comparison, the surface-based model represents V_0 by the surface S_0 at a given distance or infinity and V_1 by its outside and inside surfaces S_1 and S_2, allowing propagation and penetration to be described separately as S_{01} and S_{12} branches connecting the three nodes. Figure 1 shows the topological representation of a simple system (a display monitor) and the flow graph associated with some of its components.

1.1.2 Electromagnetic Attributes

Having translated a problem into its equivalent topology, it is then necessary to assign some electromagnetic properties (or attributes) to each component (nodes) in the topology, as well as to define the coupling paths

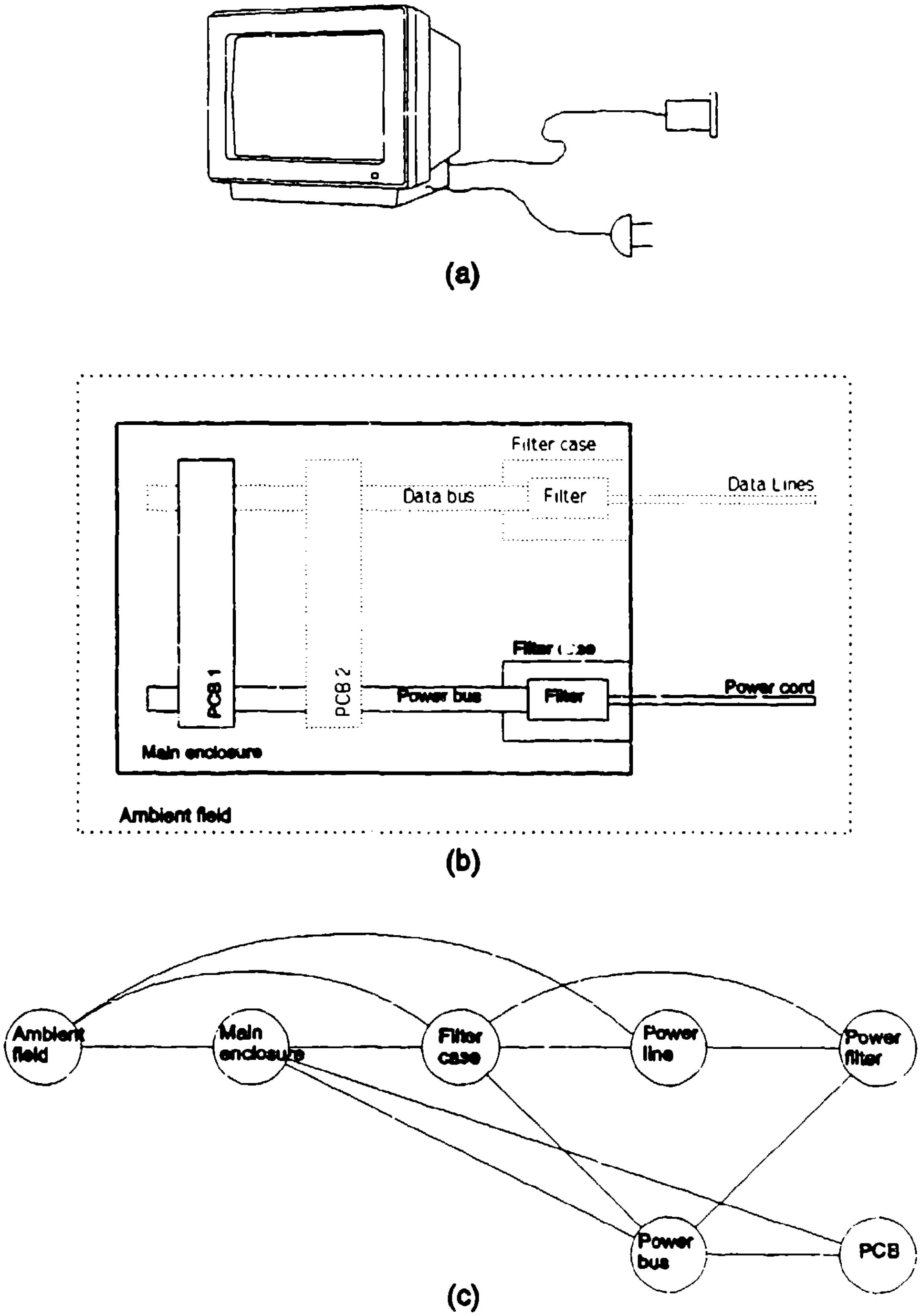

Figure 1. Electromagnetic topology of a simple system and its associated graph.

between the components (the branches of the graph).

Although, in theory, the electromagnetic attributes can be specified in either time or frequency domain, in many instances, only the frequency domain characterization is available (from calculation, measurements or manufacturer data). Furthermore, the response of a system is far easier to obtain in frequency domain, which is simply obtained from the product of the transforms as stated by the Fourier transform theory:

$$Y(\omega) = H(\omega) \cdot X(\omega) \tag{1}$$

while the equivalent solution in time domain involves a convolution.

For these reasons, all electromagnetic attributes (fields, susceptibilities and paths) are implemented into HardSys in frequency domain only.

HardSys defines three electromagnetic attributes used to describe a problem: the ambient field (AF), the shielding effectiveness (SE) and the system susceptibility (SS). These three attributes are use to compute a fourth one, the failure index, FI (also called the probability of failure, PF, in [4]). All the attributes are specified over quantized frequency ranges. Note that the different attributes need not to be specified over the same frequency ranges. However, when combined to calculate the failure index, a frequency normalization procedure takes place to convert all attributes to what is called the global frequency range. Of course, the selection of the global frequency range will have an effect on the solution.

Each node in the electromagnetic topology may be associated with one or more electromagnetic sources. These sources (referred to as the ambient field, or AF) can be expressed in terms of either field quantities (for field nodes) or circuit quantities (circuit nodes). Fields values are expressed in terms of electric field (V/m), magnetic field (A/m) or power density (W/m^2), while values at circuit nodes are in terms of voltage (V), current (A) or power (W)[1]. The ambient fields are quantized into five qualitative levels (in addition to *nil* and *unknown*) as shown in Table 1 below.

It is very important to note that the Fourier transform adds /Hz (or ·sec) to the units of the frequency domain representation of the ambient field. Note also that each range covers 40 dB or 2 decades, which is a rather large coverage.

---

[1] Unless otherwise noted, fields will be given thorough this report in terms of electric field (V/m) or voltage (V), but could be specified in other units as well.

Discrete level	Range for field nodes	Range for circuit nodes
extreme	PD > 84 $dBm/m^2/Hz$ E > 10 kV/m/Hz	P > 84 dBm/Hz V > 3.5 kV/Hz
high	PD is 44 to 84 $dBm/m^2/Hz$ E is 0.1 to 10 kV/m/Hz	P is 44 to 84 dBm/Hz V is 35 to 3500 V/Hz
medium	PD is 4 to 44 $dBm/m^2/Hz$ E is 1 to 100 V/m/Hz	P is 4 to 44 dBm/Hz V is 0.35 to 35 V/Hz
low	PD is -36 to 4 $dBm/m^2/Hz$ E is 0.01 to 1 V/m/Hz	P is -36 to 4 dBm/Hz V is 3.5 to 350 mV/Hz
very low	PD < -36 $dBm/m^2/Hz$ E < 10 mV/m/Hz	P < -36 dBm/Hz V < 3.5 mV/Hz
nil	no ambient field	no ambient field

Table 1. Definition of discrete ambient field levels

The shielding effectiveness (SE) attributes are the branches connecting the nodes of the topological graph. It is a measure of the attenuation that the ambient field will be subjected when propagated from one node to another. Four types of interactions can be defined whether we have field nodes or circuit nodes. A field-field interaction is used to model the shielding properties of enclosures (attenuation provided by the enclosure itself as well as the imperfections of the shield, such as apertures, gaskets, etc.). A circuit-circuit interaction is typical of cable connections between circuit nodes, filters, etc. The field-circuit interaction represents the coupling of electromagnetic fields into elements such as antennas, printed circuit boards (PCB's), etc. Finally, the circuit-field interaction is emission of electromagnetic fields from circuit nodes, typically from traces on a PCB. The discrete levels defined for the shielding effectiveness are shown in Table 2 below.

The system susceptibility (SS) is defined as the level of ambient field which will cause upset or damage of a component (circuit node). As with the ambient field, the system susceptibility is defined in the frequency domain in terms of a few quantized levels as shown in Table 3 below. These levels represent power (W/Hz) or power density ($W/m^2/Hz$) which will cause the upset or damage.

Finally, an assessment of the system vulnerability is done by computing the failure index (FI) of every susceptible nodes. As mentioned before, all of the attributes of every node and path are normalized to the global frequency range. This process basically takes the worst case of each attribute for each of the

Discrete level	Range for SS
excellent	SE > 100 dB
good	SE is 80 to 100 dB
fair	SE is 60 to 80 dB
not good	SE is 40 to 60 dB
poor	SE < 40 dB
nil	no shielding

Table 2. Definition of discrete shielding effectiveness levels

Discrete level	Range for field nodes	Range for circuit nodes
nil	not susceptible	not susceptible
very low	SS > 84 $dBm/m^2/Hz$	SS > 84 dBm/Hz
low	SS is 44 to 84 $dBm/m^2/Hz$	SS is 44 to 84 dBm/Hz
medium	SS is 4 to 44 $dBm/m^2/Hz$	SS is 4 to 44 dBm/Hz
high	SS is -36 to 4 $dBm/m^2/Hz$	SS is -36 to 4 dBm/Hz
extreme	SS < -36 $dBm/m^2/Hz$	SS < -36 dBm/Hz

Table 3. Definition of discrete system susceptibility levels

global frequency ranges, that is the highest field, the lowest shielding, the highest susceptibility or the worst coupling path. For the purpose of the calculation, all attributes levels are translated into small numbers. At every node, the total ambient field, denoted $\sum$AF, is calculated by taking the worst of every AF, and for every branch, the total shielding, denoted $\sum$SE, is calculated by taking the weakest of all SE. The propagated ambient field, ρAF, is calculated by subtracting the shielding effectiveness from the ambient field, ie. $\rho AF = \sum AF - \sum SE$. The system will be susceptible if the ρAF exceeds the tolerance of the most susceptible component of any given node, hence the failure index is defined as $FI = \rho AF - SS$.

1.2 IMPLEMENTATION

The version of HardSys/HardDraw evaluated in this report is the first version developed by NRC[1]. It was installed at DREO in the 1st quarter of 1992 on a SUN SPARCstation 2™.

As pointed out before, the implementation consists of two main components: HardSys and HardDraw. HardDraw runs under three existing tools: NeWS™, HyperNeWS release 1.4 (which now incorporates GoodNeWS) and Quintus Prolog release 3.1.1. NeWS (Network extensible Windowing System), under which OpenWindows 2.0 also runs, provides the standard graphics primitives on the SUN workstation. HyperNeWS/GoodNeWS is a NeWS-based interface providing objets such as buttons or sliders, as well as a drawing tool, terminal emulation, etc. HyperNeWS/GoodNeWS has been developed at the Turing Institute in Scotland, a non-profit organisation involved in research in artificial intelligence. It is mostly written in PostScript. HardSys, the expert system part, is written entirely in Prolog (Ref. [12]). An object-oriented programming environment, taken from [13], is also included.

[1] This version is referred to as version 1. Versions 2 and 3 are currently being developed at NRC and University of Western Ontario to incorporate some of the changes proposed in [5].

2.0 EVALUATION OF HARDSYS/HARDDRAW

This chapter summarizes our evaluation of HardSys/HardDraw. Comments on the implementation and on the user interactions are given, but most of the discussion will be on the algorithms used in HardSys.

2.1 IMPLEMENTATION

As mentioned in the previous chapter, HardSys (and some portion of HardDraw) are implemented in Prolog. This was a very good choice: this language is ideal for building expert systems and the version chosen (from Quintus) is very mature with very good support and is available for different platforms (DOS, UNIX, VMS, etc.).

HardDraw, on the other hand, is mostly written in PostScript and runs under HyperNeWS. The designers of HyperNeWS at the Turing Institute probably had good reasons to choose PostScript as their main programming language. Unfortunately, PostScript is a rather unusual language to learn and use. Routines written in PostScript are hard to read and thus prone to programming errors. To illustrate this, one of the HardDraw routines is shown in Figure 2. Even the designers of NeWS (under which HyperNeWS runs) at SUN strongly discourage the use of PostScript for programming. Unless a very strong expertise in PostScript exists in-house, any modification or improvement to HardDraw will be difficult to code.

Although HyperNeWS provides a complete windowing environment (multiple windows, buttons, sliders, etc.), it is based on a non-commercial product and this raises two issues. First, compatibility problems are likely to appear when the operating system or the windowing environment are updated. For instance, since the installation of this version of HardSys/HardDraw few months ago, the SUN operating system has gone through a major overhaul with the release of the new operating system Solaris™ and also, the version 3 has replaced OpenWindow 2. It is already known that the current release of HyperNeWS will not work with OpenWindow 3. The Turing Institute is an academic organization and keeping HyperNeWS up-to-date may not be their top priority Second, the windowing system implemented by HyperNeWS is not standard; it has been developed by the Turing Institute for their own needs, although they are willing to share their results. As a consequence, HyperNeWS is limited to SUN workstation platforms with the NeWS (PostScript) windowing environment and this seriously limits future implementation of HardSys/HardDraw on other platforms such as PC's. A better choice of windowing environment would have been based on the Motif standard

```
/ok_to_group { % --
    1 dict begin
    /good_group true def
    0 1 component_members length 1 sub {
        dup component_members exch get begin selected {
            group {/good_group false store} if
            1 component_members length 1 sub {
                component_members exch get begin
                selected {
                    x y w h end x y w h RectInRect not {
                        /good_group false store
                    } if
                }{
                    x y w h end x y w h rectsoverlap {
                        /good_group false store
                    } if
                } ifelse
            } for
            end exit
        }{
            end pop
        } ifelse
    } for
    good_group end
} def
```

Figure 2. Sample HardDraw routine in PostScript.

developed at the Open Software Foundation (OSF)[1]. Motif is based on the X Window specification. Both Motif and X Window are already implemented on a variety of platforms such as UNIX, VMS, IBM, etc. In addition, support of both X Window and Motif is implemented in the Quintus Prolog.

Nonetheless, HardDraw is a major piece of work, especially the portion of the code which implements the topological drawing tool and the construction of the associated flow graph.

2.2 USER INTERACTION

All functions related to user interaction are performed by HardDraw. Most user inputs are done with the use of a mouse (click on *buttons*, pop-in menus, etc.). These functions are generally well designed, but as mentioned in the previous section, their implementation in PostScript makes it difficult to customize or to create new functions.

---

[1] The Open software Foundation (OSF) is an organisation of leaders in the computer industry, promoting standardisation of software.

This version of HardSys/HardDraw lacks any hardcopy capability. This is surprising, specially considering the fact that HardDraw is written mostly in PostScript and that it should not be too difficult to add support for PostScript printers. The only alternative for now is to use the *snapshot* feature of OpenWindow to save a bitmap of a portion of the screen, but this gives copies of poor quality.

This version of HardSys/HardDraw also lacks any plotting capability. This feature would be most useful to look at the database or the results of HardSys (ambient field, susceptibility, shielding effectiveness and failure index).

2.3 MODELLING ASPECTS OF HARDSYS

This section gives comments on algorithms and models used in HardSys, basically Chapters 4 and 5 of the final report [1]. Most of these comments result in possible enhancements to HardSys discussed in Chapter 3.

2.3.1 Discussions on Discrete Levels for Electromagnetic Quantities

As described in the previous chapter, HardSys defines the ambient field and system susceptibility by using only a few discrete levels and assigning qualitative words to them (*extreme*, *high*, *medium*, *low* and *very low*, *nil* and *unknown*). These represent 40 dB intervals. Such wide intervals, will introduce errors of that order when specifying fields or susceptibilities and may produce errors in excess of 40 dB when used in expressions to calculate the shielding effectiveness or the failure index. An underestimation of the proper shielding by 40 dB may very well result in a system failure under a given threat, while an overestimation by 40 dB may prove to be very costly, such as in the case of adding extra shielding on an airborne platform.

It is customary when calculating the required shielding or protection to allow for a design margin. White [15] suggests a 20 dB default margin, although margins as low as 10 dB and as high as 30 dB are also commonly used, but may result in underdesign or overdesign. Such a design margin has little meaning when compared with the error inherent with the quantization into wide intervals.

One could define more levels, lets say 5 dB intervals, but naming them becomes useless (you would need three more qualitative terms between *medium-high* and *high*). The alternative is to provide a library of models which would be used to generate the ambient field functions (which may or may not be quantized). Although increasing the number of levels improves the definition of the flat

portion of the spectrum (0 dB/decade slope), other parts of the curve are still coarsely approximated (staircase approximation) and it will be shown later that important information can be obtained, particularly in the -20 dB/decade section.

2.3.2 Discussion on Failure Index Calculation

One apparent advantage of using wide intervals and thus using very few discrete levels, is that an ambient field may be described based on intuition. However, it is not obvious how to estimate a field in the frequency domain (given in V/m/Hz for instance) from its time domain characteristics (given in V/m). For instance, Section 4.2.2 of the final report [1] describes a double exponential model that is suitable to represent both lightning and nuclear EMP. The peak field is 50 kV/m for both the lightning (at 0.1 km) and nuclear EMP. The frequency spectrum for both fields is shown in Figure 3, along with a scale of the discrete level definitions (Figure 17 of [1], reproduced in Table 1). Therefore, a lightning EMP would be represented as:

f < 10 kHz	→	Medium
10 kHz - 400 kHz	→	Low
400 kHz - 6 MHz	→	Very low
> 6 MHz	→	nil

and similarly, a nuclear EMP would be represented as:

f < 500 kHz	→	Low
500 kHz - 70 MHz	→	Very low
> 70 MHz	→	nil

which suggests that a nuclear EMP is a relatively low threat, while it is well recognized to be a very serious threat.

In general, the time scaling property of the Fourier transform (Section 1.4 of Ref. [16]):

$$g(\alpha t) \leftrightarrow \frac{1}{|\alpha|} G(f/\alpha) \qquad (2)$$

states that, for a given waveform, the magnitude of its spectrum is proportional to its duration. Consequently, the magnitude of the spectrum (in terms of V/Hz, V/m/Hz, etc.) cannot be compared directly against a threshold for upset or damage, usually given in terms of voltage, current, power or energy, and sometimes also as a function of the duration (see Section 3.3.2 and 3.3.3 for a

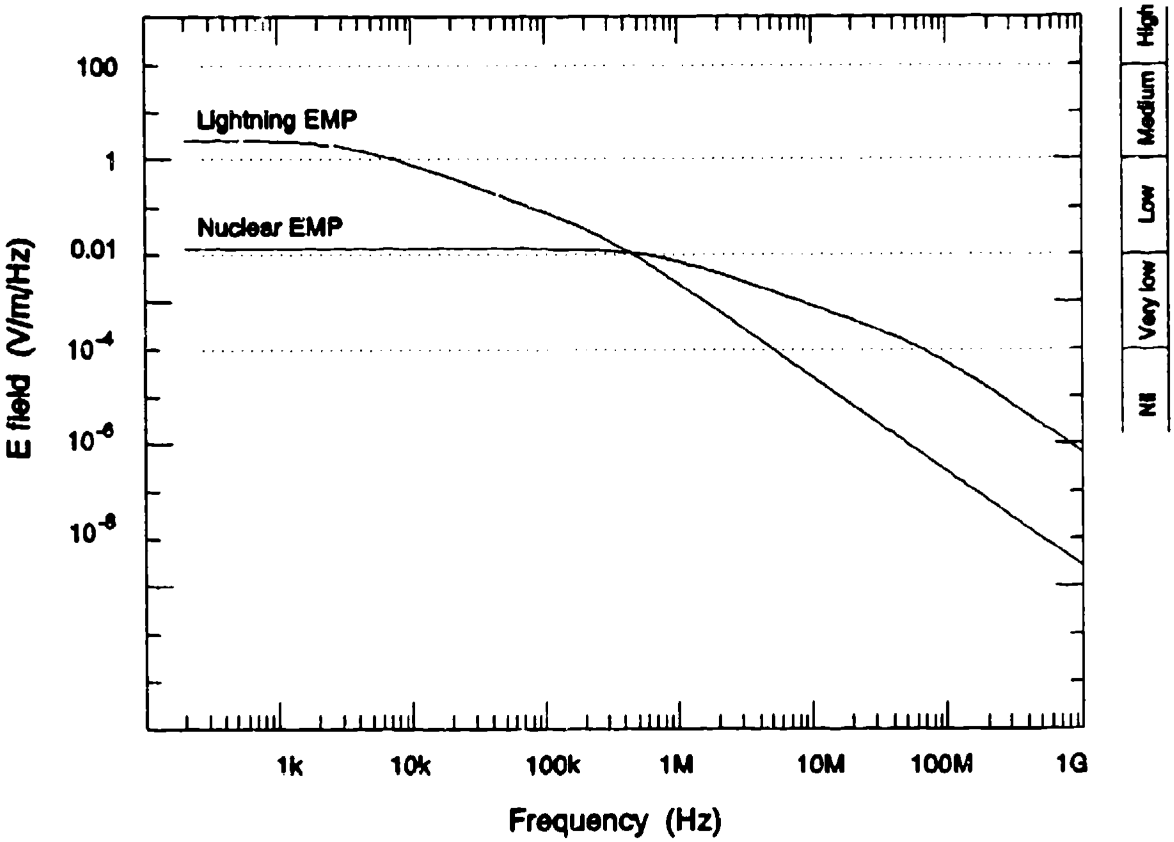

Figure 3. Frequency spectrum of lightning and nuclear EMP.

detailed discussion about thresholds). To take another example, consider the case of square pulses of 1 V amplitude and 1 μsec, 1 sec and 1 hour duration. Their spectrum[1] would have a maximum of 1 μV/Hz, 1 V/Hz and 3.6 kV/Hz respectively, corresponding to *nil*, *medium* and *extreme* levels, as defined in Table 1. All these signals may cause upset in digital logic integrated circuits. However, the algorithms described in Ref. [1][2], which is based solely on V/Hz curves, yields a likelihood of failure of *nil*, *high* and *extreme* respectively.

To further illustrate this problem, Figure 4 shows the spectrum of several waveforms (double exponential waveforms (curves 1 & 2), damped sine waves (3 & 4), gated sine waves (5 & 6) and a CW carrier (7)) which all have the same peak amplitude. This clearly indicates that a better formulation of the problem is necessary. Part of the problem lies in that the specifications for ambient field or susceptibility levels are rarely given in V/m/Hz or V/Hz (as HardSys defines ambient fields) but are usually expressed in terms of voltage, power, energy, duration, etc. A method for extracting relevant parameters such as peak value, rise time, duration, total energy, directly from the Fourier transform will be presented in the next chapter.

2.3.3 Discussion on Frequency Normalization

As described previously, all electromagnetic attributes (AF, SS and SE) are defined in the frequency domain over a set of discrete frequency ranges, which need not to be the same for all attributes. The failure index calculations developed in Chapter 3 of the final report [1] assume that all attributes have been normalized to a common range called the global frequency range. This process basically takes the worst case of all the ranges of the attribute which overlap each of the global frequency ranges. This is illustrated in Figure 5(a), which represents the addition of two ambient fields. It is obvious that a proper global range must be selected, otherwise strong levels will spread over a much wider frequency range, yielding to an overestimation of the fields (this is shown in the figure, where a 400 kHz bandwidth signal propagates thorough the whole 0.1-10 MHz band). Of course, a finer frequency range could be used, this has to be done manually by looking at each attribute obtained from the characteristics database. But the main purpose of such database is to hide from the system designer the details of the models to concentrate on system modelling; thus it would be preferable that the global frequency range be calculated automatically.

---

1 The magnitude of the spectrum of a square pulse is $V_{pk} \cdot T \cdot |\mathrm{sinc}(f \cdot T)|$ (Ref. [16]).

2 The details of these calculations are not shown here and can be found in Chapter 5 and Figures 25, 37, 40 and 41 of the final report [1].

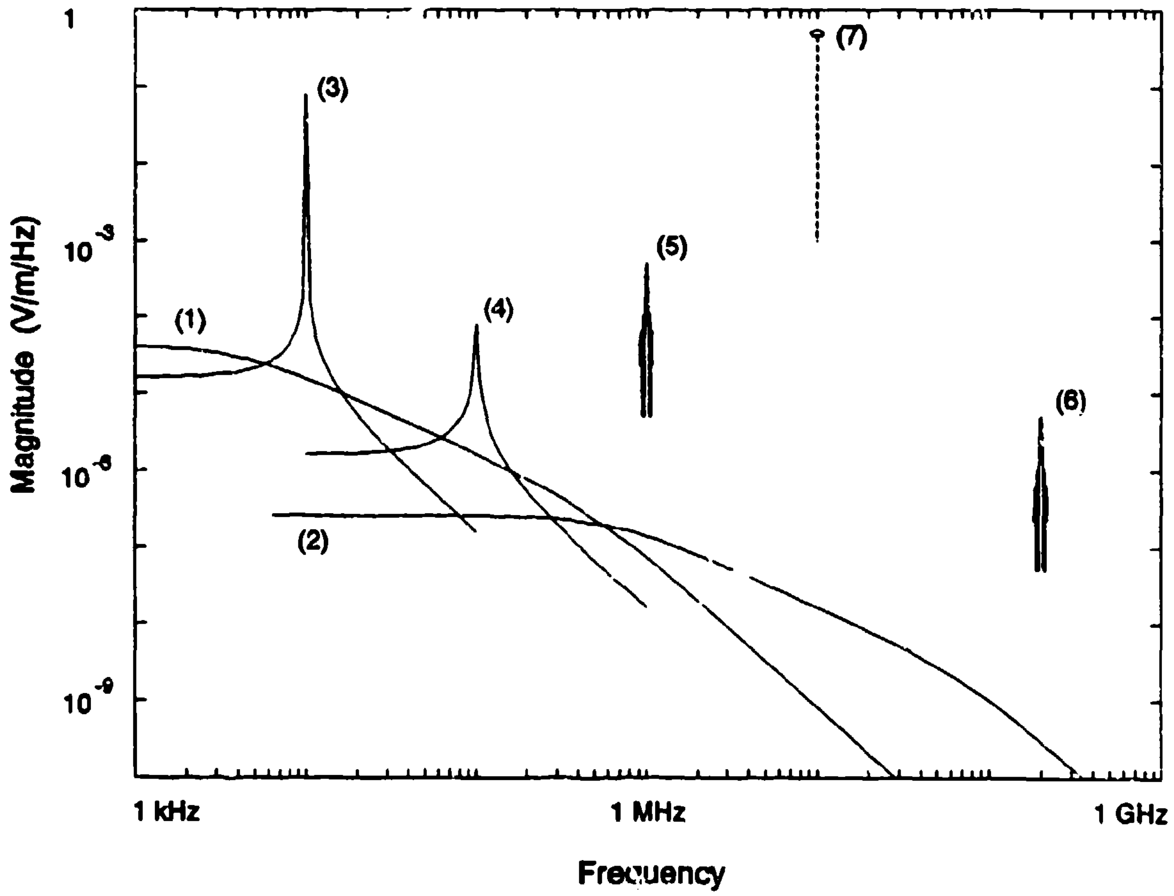

Figure 4. Frequency spectrum of several waveforms of same amplitude.

Another way to circumvent this problem is to define operators which work on attributes of different frequency ranges and thus, do not require a global frequency range. Such an algorithm is illustrated in Figure 5(b), which shows that the frequency range is automatically refined to obtain the solution.

2.3.4 Discussion on Database Representation.

HardSys/HardDraw stores in a database all the electromagnetic attributes (ambient field, system susceptibility and shielding effectiveness), along with the global frequency ranges. The definitions stored in the database are static: an attribute of a given type (AF, SS or SE) and a given name (type and subtype) is stored as a static list of frequency ranges and of qualitative strength (*low*, *medium*, *high*, etc.). This representation is easy to implement; HardDraw includes an editor to create and delete[1] attributes. However, this static representation is not very flexible and in many cases, a dynamic representation would be desirable. For instance, consider the following cases:

- A simple model may be available to calculate the attributes based on user-supplied parameters (such as length, dimensions, etc.). For instance, in a static database, it would be necessary to create a shielding effectiveness entry for each of various length (1m, 2m, 5m, 10m, etc.) and for each of the cable types; and if a particular entry is not present, it would have to be created by the end-user, who may not be expert in electromagnetic interactions.

- An ambient field may be defined with some of its parameters taking any values within a given range. For instance, an HPM threat can be defined as a very narrowband signal of a given amplitude (15 kV/m) and duration (1 μsec) with a carrier frequency anywhere in the band 500 MHz to 50 GHz. To represent that threat as the envelope of all possible spectrums is unrealistic as this would overestimate the total energy of the signal by several orders of magnitude (the Rayleigh's energy theorem in [16] shows that the total energy is proportional to the bandwidth). It would be better to have the threat *walked through* the band to identify potential susceptibilities.

- If no data or model is available for an attribute, it should be possible to extrapolate the data from another similar model. For instance, if the attributes of a 5 m monopole antenna are not known, the data for a 15 m

[1] Unfortunately, it is not possible to edit an existing attribute with the current version of HardSys.

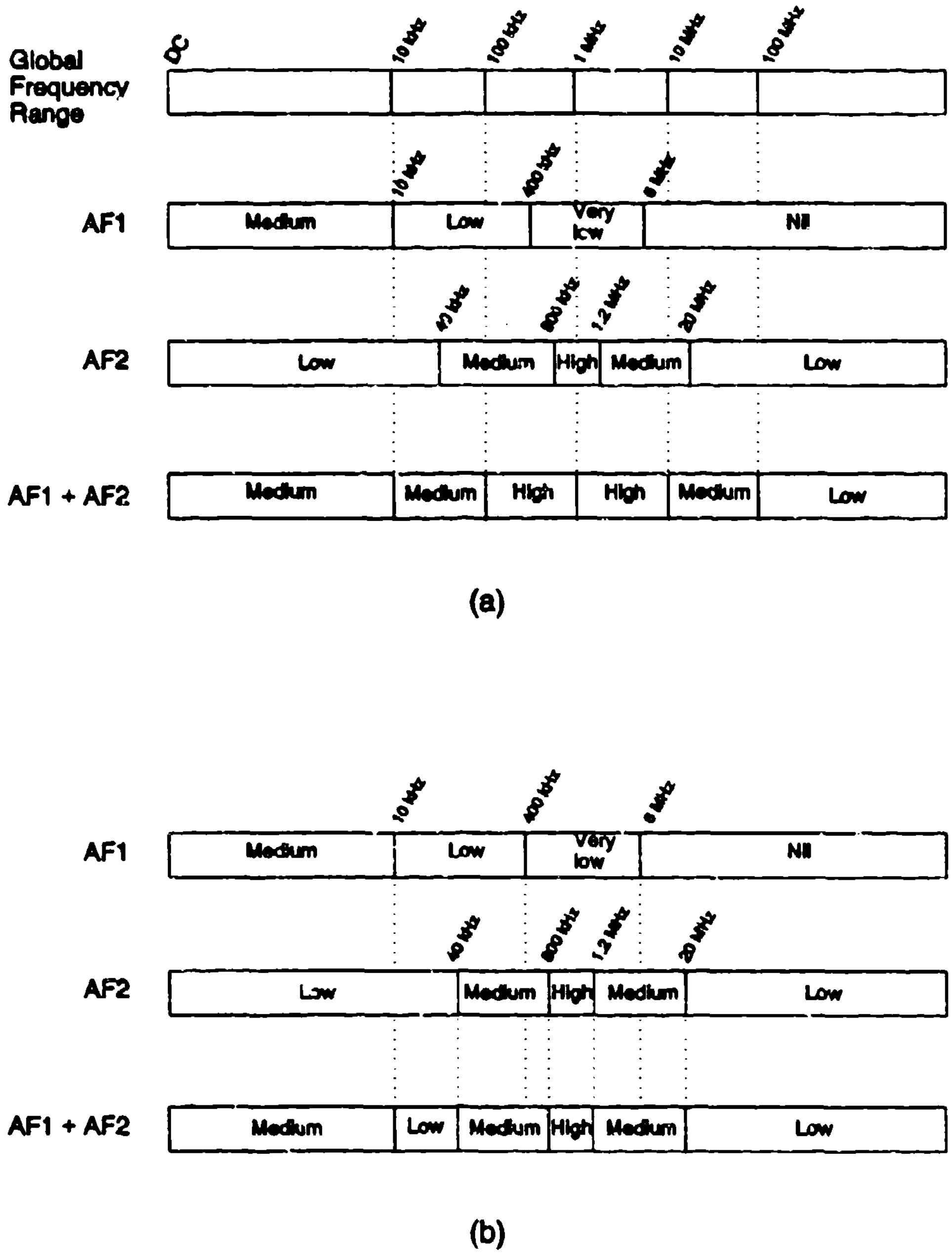

Figure 5. Global frequency range and parallel addition.

antenna found in the database could be extrapolated.

- It is also possible to estimate some attributes based on attributes of similar components. For instance, the data stored on the 74xx logic family should be used to estimate the attributes of the 74LSxx family if no data is present.

- Often, models are based on some approximation of the real world and thus are valid only over a limited range (ie. one is good at low frequencies, another at high frequencies, etc.). It is therefore possible (and necessary) to specify several models for the same attribute.

HardSys already supports many of the dynamic representations cited above. For instance, there is one dynamic model available in HardSys: the aperture of a shielded enclosure, where the SE attributes are calculated from the dimensions of the aperture. The disadvantages of dynamic representations are that the models need to be hard-coded and thus, some knowledge of Prolog is required. HardSys hard-codes the aperture model within its own code, but Prolog has already the capability to dynamically load and reload source or compiled modules.

It should also be obvious that as multiple solutions may exist (ie. for a given component, several different models may give different attributes), a mechanism to rate them is necessary. This may be in the form of a confidence factor associated with the attributes.

This version of Hardsys includes codes taken from [13] to implement object-oriented programming in Prolog. The basic idea of object-oriented programming is that the information is represented in terms of objects. The main characteristics of an object is that it binds together the definition of a data structure with some procedures. Procedures (also called methods) are activated by sending messages to the object. Objects are also instances of a class. The class defines the properties of all objects in that class. Classes themselves are organized into a hierarchy making it possible for objects of one class to inherit properties of a parent class. This inheritance concept is fundamental for object-orianted programming.

Currently, HardSys uses the object-oriented tools above only to manage its database. It does not define any hierarchy of objects (no class of objects are defined) and thus does not use inheritance to propagate attributes. This inheritance mechanism could be used to implement some of the dynamic representations discussed above.

2.3.5 Discussion on Time Domain Representation

As mentioned before, HardSys defines a problem in frequency domain only. Although this is the best approach, it has a serious limitation: only linear problems can be modelled. In hardening against electromagnetic threats, non-linear devices such as spark-gaps, varistors and Zener diodes are frequently used. The current version of HardSys has no support of non-linear devices and therefore, it is not possible to study a whole system which includes non-linear elements.

3.0 ENHANCEMENTS TO HARDSYS

This chapter presents several possible enhancements to HardSys, mostly to address the inadequacies of the models and algorithms discussed in Section 2.3. Most of these enhancements are already at various stages of development or implementation at DREO. However, none of these enhancements have been included in HardDraw (see Section 2.1).

Some enhancements have already been proposed by NRC in Ref. [5], mainly to include fuzzy arithmetic concepts. NRC has developed a new version of HardSys (version 2) which includes these concepts. However, all the enhancements proposed in this chapter are based on the first version of HardSys/HardDraw.

A completely new definition of the electromagnetic attributes is introduced, along with a set of simple rules to propagate them. A method for relating the frequency domain spectrum with significant time domain parameters such as peak value, rise time and duration is presented. A new approach to calculate the required shielding or protection, based on thresholds defined in time domain (ie. in terms of peak value, power, energy and/or duration), is described. Uncertainty concepts (fuzzy arithmetic and confidence factors) are reviewed and applied.

3.1 ELECTROMAGNETIC ATTRIBUTES REPRESENTATION

It was shown that, in order to realistically solve a problem with a 10 to 20 dB error margin, the quantization interval needs to be smaller than 5 dB, and that with such small intervals, the definition of qualitative representation becomes problematic.

A new definition similar to the Bode plot representation of a transfer function was implemented. On a Bode plot, the magnitude of a transfer function is easily obtained by locating the poles and zeros on the frequency axis: at every single pole, the slope of the curve is decreased by 20 dB/decade, and at every zero, it is increased by 20 dB/decade. The new definition consists of a list of the frequencies (corresponding to the location of poles and zeros) and the associated attribute value, given in dB. Interpolation can then be performed to obtained intermediate values. For instance, the nuclear EMP field shown in Figure 3 can be represented as[1]:

[1] All frequencies must be specified in Hz, but are shown in MHz thorough this document for clarity.

[(0,-37.7) , (0.64,-37.7) , (76,-79.2) , (300,-103.)]

which has a maximum error of 3 dB near the poles (generally, the error of the order of the pole or zero: 3 dB at single poles or zeros, 6 dB at double poles zeros, etc.). Since a model is available to describe the attribute, more frequencies can be generated to obtain a more accurate model (better than 1 dB):

[(0,-37.7) , (0.32,-38.7) , (0.64,-40.7) , (1.3,-44.7) ,
(38,-74.2) , (76,-82.2) , (150,-92.2) , (300,-103.)]

This representation of electromagnetic attributes has its advantages. There is no error associated with quantization and all parts of the curve are accurately represented: not just the flat portion. Figure 6 below shows the simple program which generated the transfer function of a nuclear EMP shown above.

```
source(emp,Params,TF) :-
    emp(Params,fc,(Fc1,Fc2)),
    Fc2b is 2*Fc2,

    emp(Params,mag_dB,(0,E0)),       emp(Params,mag_dB,(Fc1,E1)),
    emp(Params,mag_dB,(Fc2,E2)),     emp(Params,mag_dB,(Fc2b,E2b)),

    TF = [ (0,E0;1), (Fc1,E1;1), (Fc2,E2;1), (Fc2b,E2b;0.8) ].

emp((Alpha,Beta,AV),mag_dB,(F,MAG)) :-
    pi(2*F,W), S=(0,W),
    MAG is_dB AV * (1/(Alpha+S) - 1/(Beta+S)).

emp((Alpha,Beta,_),fc,(Fc1,Fc2)) :-
    Fc1 is Alpha/6.2832, Fc2 is Beta/6.2832.
```

Figure 6. Simple models to generate the transfer function of a double exponential waveform.

Appendix A shows (in part) the Prolog implementation of this representation. The electromagnetic attributes (referred to as transfer functions in the program) are represented as a list where each element specifies a frequency (f) and a value in dB (v), and may but additionally include the logarithm of the frequency (for optimisation purpose) and a confidence factor[1]. For simplicity and compactness, the following notation is used to specify each element:

(f[/log(f)],v[;cf])

---

[1] Confidence factors will be discussed in later sections.

where cf is the confidence factor which defaults to 1 and [ ] denotes optional parameters. Note that the use of parentheses is required. The examples below show valid notation for the same specification:

(1000,63) (1000/3,63) (1000,63;1) (1000/3,63;1)

The predicate (Prolog statement) tf_evaluate performs interpolation on a transfer function for a given frequency. An extrapolation is performed if the frequency is outside the range of the transfer function. The predicate tf_propagate propagates the electromagnetic attributes. Basically, it takes two transfer functions as inputs and calls an external function[1] (intrinsic or user-defined) which computes the value of each element of a new transfer function. For instance, to take the approach of HardSys which returns the worst case, calling tf_evaluate with the external function max (return the maximum of two numbers) would be similar to the parallel addition algorithm (addition of ambient fields) defined in Section 4.2.3 of the final report [1], and similarly, calling tf_evaluate with the external function min (return the minimum of two numbers) would calculate the worst case shielding path. Propagating an ambient field through a shield is simply done by subtraction. The following example[2] adds two ambient fields together (an EMP waveform and a narrowband signal around 14-16 MHz), taking the worst case:

```
TF1= [ (0,-37) , (0.64,-37) , (76,-79) , (300,-103) ],
TF2= [ (12,-60) , (14,-30) , (16,-30) , (18,-60) ],
tf_propagate(TF1,TF2,TFR,(max,_)).

TFR= [ (0,-37) , (0.64,-37) , (12,-48) , (14,-30) , (16,-30) , (18,-55) ,
       (76,-79) , (300,-103) ]
```

where it should be noted that tf_propagate automatically expands the transfer functions to cover all the frequency ranges. Therefore, the use of a global frequency range is not necessary, although it could be used for quick calculations.

Another approach to add ambient fields and shields in parallel is to convert the values in dB back to numbers, perform the operation and then convert the result to dB. The obvious drawback of this method is that the logarithm and power_of functions have to be evaluated repeatedly. Figure 7 below shows an

---

1 There are actually two external functions supplied: one which computes the value (in dB), and one which computes the new confidence factor.

2 The confidence factor is not shown here (CF=1 assumed). The unspecified argument of tf_propagate must be specified but is not shown here.

implementation of two predicates, dB_add and dB_parallel, which are used to add in parallel two ambient fields and two shielding effectiveness respectively. This piecewise approximation has a maximum error of 0.6 dB, which is quite acceptable.

```
dB_mul(X,Y,Z) :-  Z is X+Y.                        % Multiply
dB_div(X,Y,Z) :-  Z is X-Y.                        % Divide

dB_add(X,Y,Z) :-                                   % Add 2 dB numbers
    ( X>Y -> X1=X,X2=Y ; X1=Y,X2=X ),
    T is X1-X2,
    ( T>=20 -> Z=X1 ; Z is 0.7*T+6+X2 ).

dB_parallel(X,Y,Z) :-                              % Add 2 dB numbers
    ( X>Y -> X1=X,X2=Y ; X1=Y,X2=X ),              % in parallel
    T is X1-X2,
    ( T>=20 -> Z=X2 ; Z is X1-(0.7*T+6) ).
```

Figure 7. Definition of operators for dB values.

3.2 UNCERTAINTY AND FUZZY ARITHMETIC CONCEPTS

Expert systems must be able to draw conclusions based on available information. However, in most cases that information is usually not exact or more than one solution can be obtained. The system must deal with this uncertainty appropriately. Two approaches will be discussed: the confidence factor and the fuzzy arithmetic.

3.2.1 Confidence Factors

The confidence factor is a measure of the relative strength of a quantity. It is usually a number between 0 and 1[1] where a value of 1 represents certainty. In Prolog terms, the confidence factor can be applied to the antecedents (or arguments), to a rule and to the conclusion (or the result). Various approaches have been used to combine the various confidence factors to calculate the confidence factor of the solution.

The simplest method is to take the minimum confidence factor of all the antecedents and to multiply it with the confidence factor assigned to the rule.

1 A range between -1 and 1 is also commonly used, where negative confidence represent the likelihood that a conclusion is NOT true.

For multiple conclusions, the maximum confidence factor of all the conclusions is taken. This method favours the strongest rule and the weakest conclusion.

Another method, based on Bayesian probability uses the confidence factor to represent a percentage of accuracy rather than relative strength. The confidence factor of a conclusion is obtained by taking the product of the confidence factors of all antecedents, multiplied with the confidence factor of the rule. The confidence factor of multiple conclusions is accumulated as $\sum CF = \sum CF + CF \cdot (1-\sum CF)$.

It is not always clear which of the methods is the best suited for a given application, and although the actual value of the confidence factors may not be accurate, is has been shown that the ranking of the conclusions is accurate.

A problem encountered with the methods above is that a solution will be given a low confidence factor if one of its antecedents is imprecise, even if it is actually a negligible quantity. Consider for instance adding the two numbers 1000 and 5, which have a confidence factor of 0.95 and 0.6 respectively. The two methods above would give the answer 1005 and a confidence factor of 0.6 and 0.57 respectively, but a value of 0.94 would be a closer estimation.

A better method would be to have the confidence factor represent a measure of the accuracy and to write a rule to calculate the confidence factor for each operator (addition, substraction, etc.). For instance, the confidence factor could be a measure of the error in dB where the values of 1, .9, .8, ..., .1, 0 would represent an error of 0, 6, 12, 54 and <60 dB respectively. Appendix B shows a Prolog implementation of confidence factor arithmetic. Quantities are specified with the notation (v;cf) where cf is the confidence factor. Note that the use of parentheses is required, but may be omitted if the confidence factor is not specified (cf=1 is assumed). Addition, multiplication and division operators and their decibel counterparts are defined. Parallel addition of decibel quantities is also implemented. Operators with decibel quantities are based on approximation for efficiency. Confidence factors arithmetic can also be used with transfer functions (as defined in Section 3.1). For instance, adding two transfer functions is simply done as:

```
tf_propagate(TF1,TF2,TFR,(cf,+@)).
```

where +@ represents the addition of decibel quantities, but could also be *@, /@, +//@ for multiplication, division and parallel addition of decibel quantities respectively. The following definition and statement can be used to return the best (in each frequency range) of two transfer functions:

```
tf_best((FV1;CF1),(FV2;CF2),(FV3;CF3)) :-
    CF1 >= CF2 -> FV3=FV1, CF3=CF1 ; FV3=FV2, CF3=CF2.

tf_propagate(TF1,TF2,TFR,tf_best).
```

3.2.2 Fuzzy Arithmetic

The concept of fuzzy arithmetic was first introduced in 1965 and has been subject to considerable research and applications in the past decade. Fuzzy arithmetic can be considered as an extension of the concept of the interval of confidence. It should not be confused with fuzzy logic which is a definition of boolean operations on fuzzy sets, ie. sets that allows its members to have different grades membership (or partial membership). A good introduction to fuzzy arithmetic can be found in [19].

A fuzzy number A (or fuzzy subset) can be defined by its membership function (or level of presumption) $\mu_A(x)$. Contrary to conventional set theory where $\mu_A(x)$ may only take the values 0 and 1, ie. an element either belong to, or does not belong to A, this function may take any value in the range [0,1]. The $\mu_A(x)$ function is said to be normalised when its maximum value is 1. The interval of confidence for a given level of presumption α is noted $A_\alpha = [A_1^{(\alpha)}, A_2^{(\alpha)}]$ and its relation to the level of presumption is shown on Figure 8 (a). The function $\mu_A(x)$ of a fuzzy number A can be of any shape and except for few very simple cases and only for few operators, its shape is generally not preserved. For this reason, and to have an efficient computer implementation, a simple $\mu_A(x)$ function is usually chosen, which can be then defined by using only few parameters. One such definitions often used is the trapezoidal fuzzy number, as shown on Figure 8 (b), where the fuzzy number A (normalised) can be fully described as (a_1, a_2, a_3, a_4).

The addition of fuzzy numbers is done by adding the two intervals of confidence, but level by level. This can be written as ([19]):

$$\begin{aligned} A_\alpha \,(+)\, B_\alpha &= [a_1^{(\alpha)}, a_2^{(\alpha)}] \,(+)\, [b_1^{(\alpha)}, b_2^{(\alpha)}] \\ &= [a_1^{(\alpha)} + b_1^{(\alpha)}], [a_2^{(\alpha)} + b_2^{(\alpha)}] \end{aligned} \tag{3}$$

where (+) represents the fuzzy operator. It can be proven that the addition of two trapezoidal fuzzy numbers results also in a trapezoidal fuzzy number, as illustrated on Figure 9 (a). Similarly, the subtraction of two trapezoidal fuzzy numbers yields to a trapezoidal fuzzy number. Thus, for fuzzy addition (and

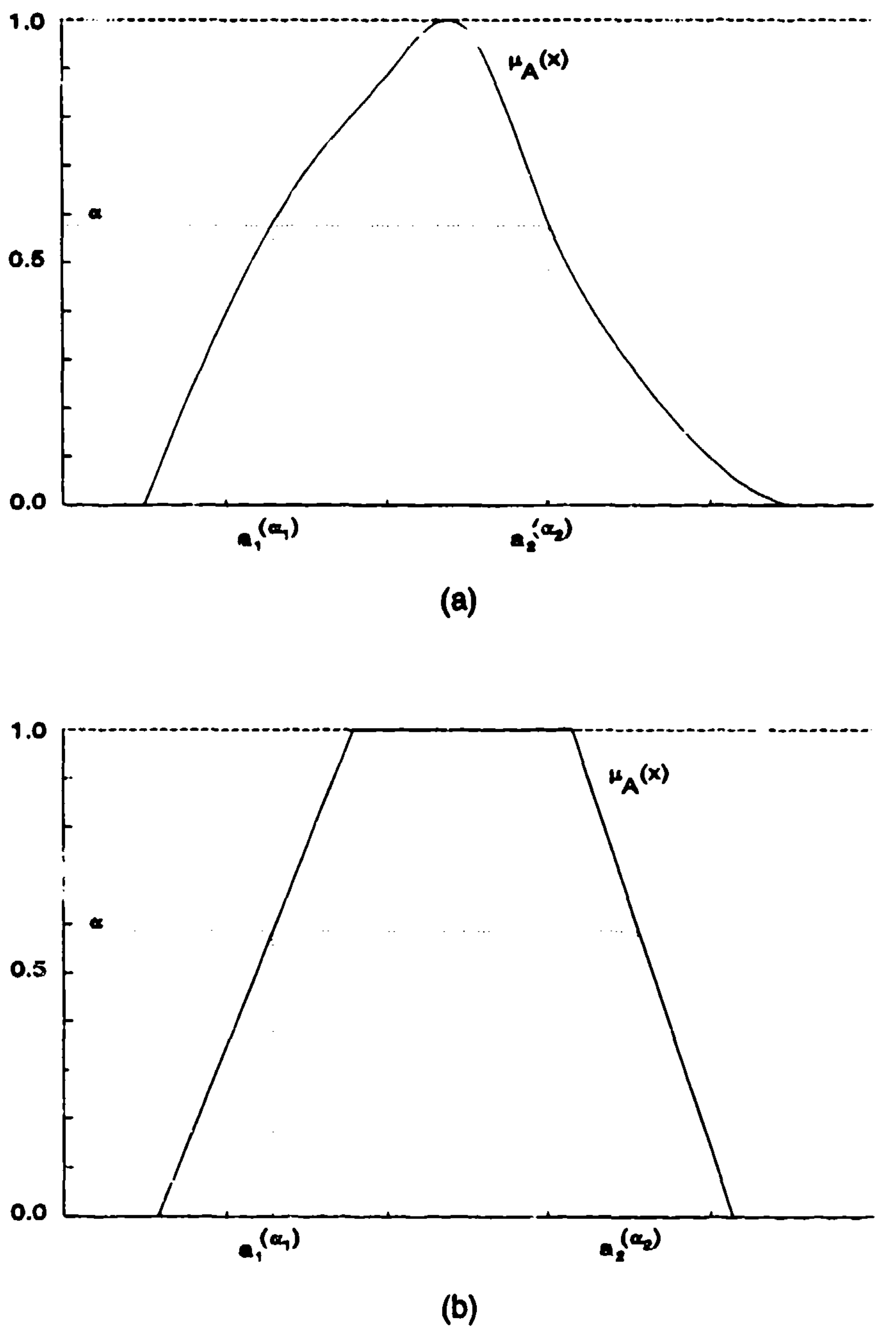

Figure 8. Definition of fuzzy numbers.

subtraction), we can write:

$$A\ (+)\ B = (a_1, a_2, a_3, a_4)\ (+)\ (b_1, b_2, b_3, b_4) = (a_1+b_1\,,\, a_2+b_2\,,\, a_3+b_3\,,\, a_4+b_4) \quad (4)$$

Fuzzy multiplication and division are defined by a relation similar to Equation (3) above. What is most important to note is that multiplication or division of trapezoidal fuzzy numbers does not yield a trapezoidal fuzzy number, as illustrated in Figure 9 (b)[1]. In this case, the sloped portions of the curve (segments $[c_1, c_2]$ and $[c_3, c_4]$) are actually square root functions, which get more complicated if this result is further used in subsequent fuzzy operations. However, as the curvature of sloped portions is not pronounced, it is legitimate to approximate it as a trapezoidal fuzzy number and describe it using the (c_1, c_2, c_3, c_4) notation, and use the relation:

$$A\ (\bullet)\ B = (a_1, a_2, a_3, a_4)\ (\bullet)\ (b_1, b_2, b_3, b_4) = (a_1 \cdot b_1\,,\, a_2 \cdot b_2\,,\, a_3 \cdot b_3\,,\, a_4 \cdot b_4) \quad (5)$$

Several techniques also exist to convert a fuzzy number into a *crisp* number (defuzzification). One of these techniques uses the center of gravity of the area under the $\mu_A(x)$ curve, also called the centroid, to calculate the *center* of the interval of confidence. It is also possible to force a bias toward under- or over-estimation.

Appendix C shows a Prolog implementation of fuzzy arithmetic, using the same infix notation used for regular mathematical expressions. A formulation similar to Equations (4) and (5) is extended for others monadic operators such as *sqrt(x)*, and diadic operators, such as *power_of(x,y)*, *min(x,y)*, *max(x,y)*, although this is an approximation as the result is not a trapezoidal fuzzy number. Note that the algorithm shown in Figure 7 is also included to provide efficient calculation of fuzzy decibel quantities.

It can also be demonstrated that the addition or multiplication of fuzzy numbers is commutative and associative; however, their inverse is not symmetric, ie. $A\ (+)\ A^-$ is not equal to zero and $A\ (\bullet)\ A^{-1}$ is not equal to 1. In general, great care should be exercise when coding an algorithm. For instance, consider the case of parallel addition, which can be coded using one of two mathematically identical expressions:

---

[1] What is shown here is the multiplication of two positive fuzzy numbers. The case of negative numbers is more complex and is not treated here.

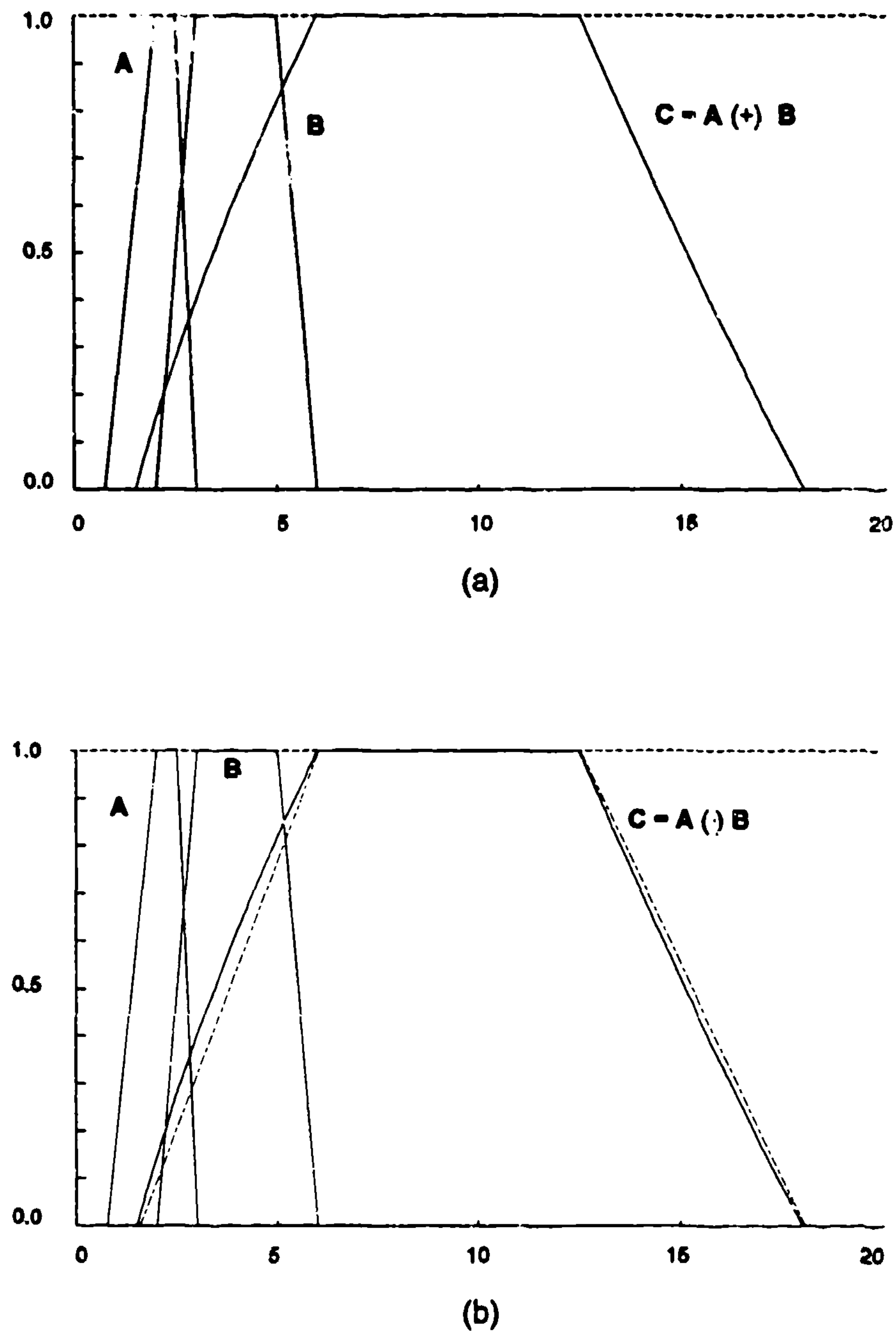

Figure 9. Addition and multiplication of two fuzzy numbers.

$$C = A /\!/ B = \frac{1}{1/A + 1/B} = \frac{A \cdot B}{A + B} \qquad (6)$$

but one of which will tend to exaggerate the fuzziness of the result in the cases of a negligible quantity with large interval of confidence. For instance, if we take A= 10 and B= 1000, with an error of 1% and 20% respactively:

A = (9.9,9.95,10.05,10.1)
B = (800,900,1100,1200)

the expression C -: A*B/(A+B) yields to:

C = (6.5,8.1,12.1,15.0)

which shows a rather large interval of confidence, while the better expression C -: 1/ (1/A + 1/B) gives:

C = (9.78,9.84,9.96,10.02)

which is more accurate (in particular, the result is smaller that the smallest of A and B, as expected).

The example below uses fuzzy arithmetic to estimate some time domain parameters, such as the rise time, pulse width and peak amplitude, from parameters obtained from the frequency domain representation, in this case, the peak magnitude and the two cut-off frequencies ($\omega = \alpha$ and β). The exact analytical solution is hard to obtain, however a good approximation (5%) can be derived[1]:

$$t_r = \frac{2.2}{2.8\alpha + \beta} \qquad (7)$$

$$t_w = \frac{0.7}{\alpha} + \frac{1}{\beta - 0.4\alpha} \qquad (8)$$

and if we choose α and β to have a nominal value of $4 \cdot 10^6$ and $4.76 \cdot 10^8$ (the standard nuclear EMP waveform) and an error of 10 and 50% respectively:

Alpha= (3.6e6,3.8e6,4.2e6,4.4e6)
Beta= (2.4e8,3.6e8,6.0e8,7.1e8)

[1] These equations are a little more complex than those given in [17], but their accuracy remains good as α and β get closer to each other, ie. it is not assumed that $\beta >> \alpha$.

and we can use simple Prolog expressions to calculate the rise time and pulse width:

```
Tr ~: error( 2.2/ (Beta + 2.8*Alpha), 5 )
Tw ~: error( 0.7/Alpha + 1/(Beta-0.4*Alpha), 5 )
```

where the error predicate assigns an error of 5% to the expression. These fuzzy expressions yield to:

```
Tr= (2.9e-9,3.5e-9,6.1e-9,9.3e-9)
Tw= (1.5e-7,1.6e-7,1.9e-7,2.1e-7)
```

which shows that the estimation of the rise time is quite inaccurate (due to the large uncertainty of β) and that the estimation of the pulse width is relatively accurate (mostly due to the inaccuracy of α, but little effect of the large error of β). These fuzzy quantities can be converted back to crisp numbers with the defuzzy predicate:

```
defuzzy(Tr,0.33,Trn)  →  Trn= 4.7 ns
defuzzy(Tw,0.66,Twn)  →  Twn= 186 ns
```

which shows here a bias toward the worst case, ie. faster rise time and longer duration.

3.3 FAILURE INDEX AND SHIELDING EFFECTIVENESS CALCULATIONS

Solving an electromagnetic problem, whether with topological decomposition or other methods, is two-fold: analysis and design. The analysis process takes a given problem (known excitations, transfer functions, shields, etc.), computes the outputs and compares them with thresholds or limits. It was discussed in Section 2.3.2 that the frequency domain representation cannot be used directly to compare electromagnetic quantities against thresholds. The inverse transform could be computed to obtain a time domain response, but its calculation is complicated by the fact that the phase information is usually not available. The design process takes one or more given excitations (from standards or obtained from analysis or numerical simulations), transfer functions, partial shields and uses the known thresholds to compute the required additional shielding. One could compute the time domain response of the system with the method described above, compare it with the given thresholds and add shielding to bring it below the thresholds. A major disadvantage of this method is that the calculated additional shielding is often overestimated as it applies to the whole frequency band although shielding or filtering over only a portion of the band may be sufficient.

3.3.1 Alternate Representation of Electromagnetic Attributes

A new method for both analysis and design calculations is presented in this section. This method is suitable for both graphical calculations (similar to Bode plots) and to computer implementations (using more elaborated models for better accuracy). Electromagnetic interactions (fields and coupling interactions) are divided into two general classes: wideband (or broadband) and narrowband. Impulses such as EMP are wideband; damped sinusoid and the gated sinusoid are two examples of narrowband signals. Wideband fields are defined with the representation described in Section 3.1. An additional attribute may be used to specify the bandwidth, *wideband* in this case. Narrowband fields are represented as a discrete quantity in the frequency domain, at the center frequency with an amplitude corresponding to the peak value in time domain, given in dB for consistency, along with an additional attributes to specify the bandwidth (bw), in Hertz. Alternatively, it can be specified using the resonance factor (Q), the damping factor (z) or the duration (T). They are all related to each other as:

$$\frac{BW}{f_r} = \frac{1}{Q} \quad , \quad Q = \frac{1}{2z\sqrt{1-z^2}} \quad , \quad BW = \frac{1}{T} \qquad (9)$$

but it should be noted that the last expression is not very accurate (±10 dB error) and that it should not be used to compute the bandwidth. Narrowband signals can be expressed as an extended transfer function consisting of two elements: the center frequency and peak value (or a transfer function as discussed below) and a list of attributes. For instance, a pulse CW signal of 1 kV at 500 MHz and of 1 μsec duration can be represented as:

```
( (500,60) , (bw=2.6,t=1e-6) )
```

where the bandwidth[1] is estimated from the known Fourier transform of the signal. The following Prolog statement can be used to extract any of the components of the transfer function:

```
XTF=( (Fc,V) , ATTR ), member(bw=BW,ATTR).
```

It is also possible to extend this definition for cases where the narrowband signal covers a range of frequencies. This is simply done by specifying a complete transfer function (as defined in Section 3.1) instead of

---

[1] Bandwidth must be specified in Hz, but is shown in MHz thorough this document.

the single frequency (f_c,v). For instance, an HPM threat consisting of a CW signal between 500 MHz to 50 GHz (with some derating down to 50 MHz and up to 500 GHz), of 1 μsec duration and 15 kV/m amplitude can be represented as:

([(50,63.5) , (500,83.5) , (50e3,83.5) , (500e3,63.5)] ,
[bw=2.6])

and the damped sinusoidal waveforms specified by the CS10 and CS11 requirements for current injection of MIL-STD-461C (Ref. [18]) can be represented:

([0.01,-16) , (0.63,20) , (10,20), (100,0)] , [q=15])

where it should be noted that the resonance factor is specified, and therefore the bandwidth will be dependant of the center frequency as stated in Equation (9).

The same representation is also used to describe transfer functions (shielding effectiveness). Wideband filters use the same representation defined in Section 3.1, with the additional *wideband* attribute. Narrowband filters are represented as a discrete quantity at the resonant frequency with an amplitude corresponding to the peak magnitude (in dB), ie. the magnitude at that frequency, along with one or more of the attributes of bandwidth, Q, z or T. Narrowband filters may also be defined using a range of frequencies as described above. This representation may be used for instance to describe the characteristics of a receiver which may be tuned at different frequencies within a range.

3.3.2 Relating the Peak Value with the Frequency Domain

It has already been discussed (Section 1.1.2) that electromagnetic problems are usually worked out in frequency domain. It has been also discussed that thresholds for upset or damage are usually a function of a time domain parameter, such as peak value, power, energy and/or duration (Section 2.3.2). This section describes a simple algorithm to relate some time domain parameters with the frequency domain spectrum (only the amplitude is required).

With the representation defined in the previous section, the peak value in time domain of narrowband signals is stored and obtained directly and it will be shown in Section 3.3.4 how it is propagated. To obtain the peak value of a wideband signal, its transform is multiplied by ω, and it can be proven that the maximum of the function $|H(\omega)|\cdot\omega$ is a very good approximation of the peak value of the signal in time domain. Conversely, any wideband function whose product $|H(\omega)|\cdot\omega$ does not exceed a threshold u_{th}, specified in time domain units such as

volt, amp, etc., will not exceed that threshold in time domain. Alternatively, the function $|H(\omega)|$ could be checked against the function μ_{th}/ω. The threshold need not be a constant, but may vary with frequency. For instance, the susceptibility threshold (for upset) of various logic families may be modelled (Section 2.3.2 of Ref. [15]) as a constant up to a given frequency, related to the bandwidth of the family, (0.4 V and 32 MHz for standard TTL family), and be increased with frequency (with a slope of 20 dB/decade) for up to 60 dB, as shown in Figure 10.

To prove the relation between the peak value in time domain and the $|H(\omega)|\cdot\omega$ function, consider the simple double exponential waveform, which is well representative of wideband signals. Its spectrum is given by:

$$H(s) = A\cdot\left\{\frac{1}{s+\alpha}-\frac{1}{s+\beta}\right\} = \frac{A(\beta-\alpha)}{(s+\alpha)(s+\beta)} \tag{10}$$

and its magnitude given by:

$$|H(\omega)| = \frac{A(\beta-\alpha)}{\sqrt{(\omega^2+\alpha^2)(\omega^2+\beta^2)}} \tag{11}$$

and, by taking the derivative of $|H(\omega)|\cdot\omega$, we can find its maximum, which occurs at $\omega=\sqrt{\alpha\beta}$ and given by:

$$\max(|H(\omega)|\cdot\omega) = A\cdot\frac{(\beta-\alpha)}{(\beta+\alpha)} \tag{12}$$

This relation is also shown in Figure 11 as the error (in dB) in estimating the peak value in time domain versus the ratio β/α. The agreement is quite good as the error is less than 1 dB for $\beta>>\alpha$ and still good (error less than 3 dB) for $\alpha\approx\beta$.

The same development can be made, either analytically or numerically, for other functions H(s). It can be generalized that the accuracy of the above approximation depends mostly on the ratio of the two cutoff frequencies delimiting the top portion of the $|H(\omega)|\cdot\omega$ function (delimiting the left and right portions, as discussed below), as given by Figure 11. In particular, it can be shown that a steeper slope (more than +20 dB/decade) on the left portion of $|H(\omega)|\cdot\omega$ (corresponding to a positive slope of $H(\omega)$ for $\omega<\alpha$) as the result of one or more zeros and/or poles at $\omega<\alpha$, will affect mostly the late portion of the signal, causing what is often called the droop effect, but will not affect the peak value or rise time significantly. The same can also be said for slope less than +20 dB/decade on the left portion, where at the limit, a slope of 0 would

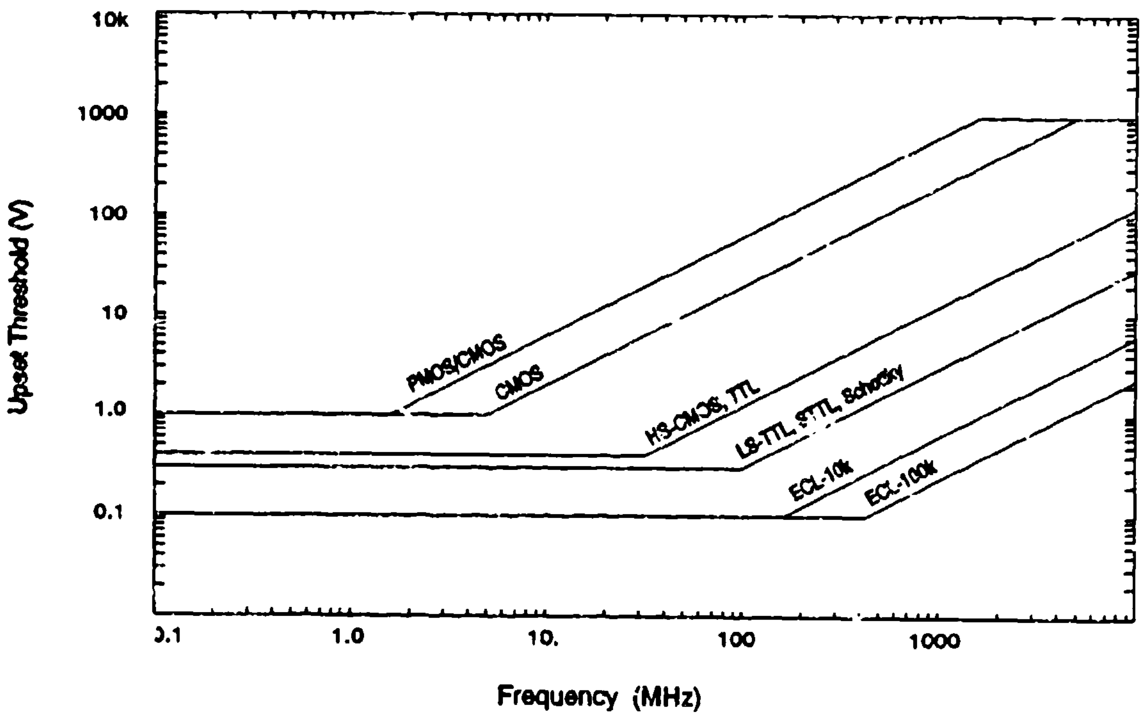

Figure 10. Upset threshold for various logic families.

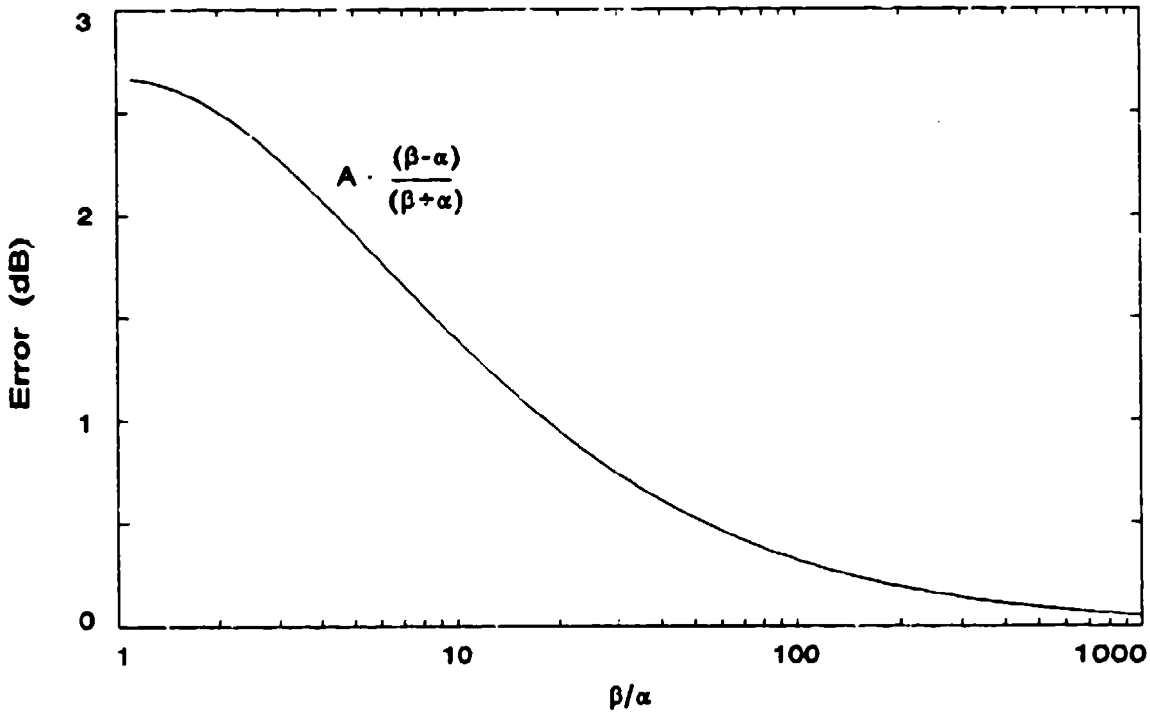

Figure 11. Error in the relation $|H(\omega)|\cdot\omega$ for estimating the peak value of a double exponential signal.

make it impossible to distinguish the left portion from the top portion. Consider for instance the function $H(\omega)=1/s$, corresponding to the unit step function $u(t)$. Its product $|H(\omega)|\cdot\omega$ can still be related to the amplitude of $h(t)$. Of course, estimation about the duration or fall time should be taken very cautiously; either assume infinite duration as for $u(t)$, or presume not enough is known about the lower frequencies. Similarly, a steeper slope on the right portion of $|H(\omega)|\cdot\omega$ as the result of poles at $\omega>\beta$ will mostly alter the shape of the rising portion of the signal, but will not affect the peak value and duration significantly. Also, for the pole at β missing where it is not possible to distinguish the right portion from the top portion, the product still gives a good estimation of the peak value and duration, but it can only be assumed that the rise time is smaller than the reciprocal of the highest frequency. The fuzzy arithmetic concepts introduced in Section 3.2.2 may be used to deal with these uncertainties.

To illustrate the use of this method, consider the following transfer function with 5 poles and 2 zeros :

$$H(s) = K s \cdot \frac{1}{s+1} \cdot \frac{1}{s+5} \cdot s+60 \cdot \frac{1}{s+200} \cdot \frac{1}{s+1000} \cdot \frac{1}{s+5000} \tag{13}$$

which is shown in Figure 12 (top, dotted curve), along with the $|H(\omega)|\cdot\omega$ product. The corresponding time domain function $h(t)$ is shown for reference in Figure 12 (bottom). The error between the peak value of $h(t)$ and the maximum of the $|H(\omega)|\cdot\omega$ product (8.7 and 10 respectively[1]) is about 1.2 dB.

Furthermore, the $|H(\omega)|\cdot\omega$ product identifies the portion of the spectrum which contributes the most to the time domain function and can be used to find the two cutoff frequencies α and β (the -3 dB points below and above the frequency of the maximum). Equations (7) and (8) can then be used (very cautiously) to estimate the rise time and width of the pulse. In the above example, the measured t_r and t_w are 1.3 and 10 ms respectively, while the estimated values (for α=130 and β=1270 rad/sec, as shown in Figure 12) are 1.35 and 6 ms respectively. The agreement for t_r is quite good, while the error in t_w (~4 dB) as expected due to the presence of the nearby zero which boosts the lower frequencies, thereby increasing the duration.

This method is not only effective in obtaining useful parameters from the frequency spectrum, but it can also be used to calculate the minimum additional shielding required to meet a given threshold (μ_{th}). By working with the $|H(\omega)|\cdot\omega$

---

1 Units are not shown for this generic problem, but could be V or V/m in time domain and V/Hz or V/m/Hz in frequency domain.

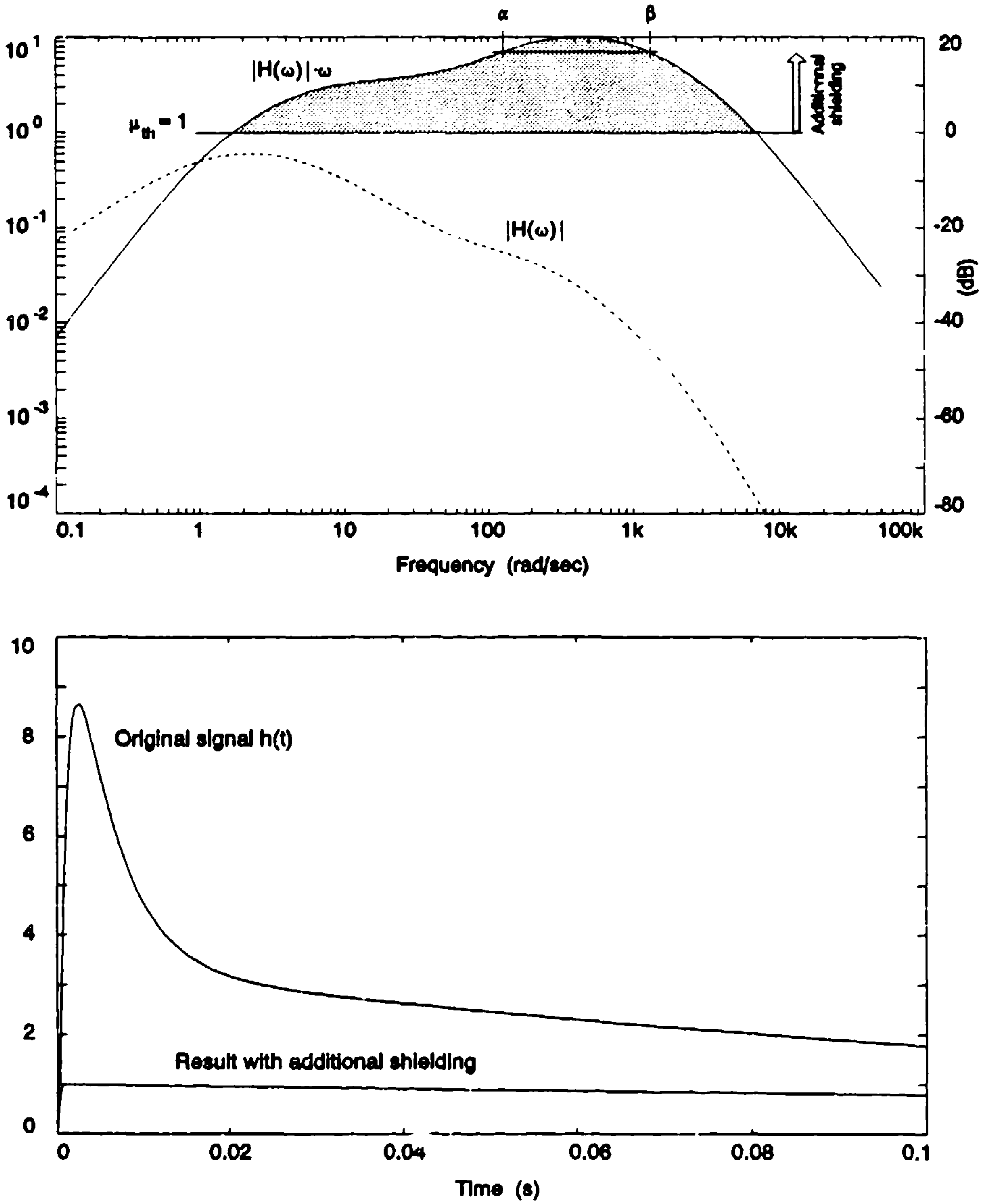

Figure 12. Comparison of the $|H(\omega)|$ and $|H(\omega)|\cdot\omega$ curves for a 5-poles, 2-zeros function, and comparison of the peak of the $|H(\omega)|\cdot\omega$ and time domain curves.

curve (in dB), the additional shielding is then found with the very simple expression:

$$\max(0, |H(\omega)| \cdot \omega - \mu_{th}) \tag{14}$$

as shown in Figure 12 (top) where the shaded area corresponds to the required additional shielding for a threshold $\mu_{th}=1$. Figure 12 (bottom) also shows the effect of the additional shielding on $h(t)$.

In conclusion, this method can accurately relate some time domain parameters (peak value, rise time, duration, etc.) with the magnitude of the spectrum. However, it is based on approximation and it has been proven valid for some types (shapes) of spectrum only and if used with other types of spectrum, those results should be questioned or verified by other methods. In summary:

> For wideband and marginally wideband signals (ie. non-resonating), the peak value in time domain can be obtained by taking the maximum of the $|H(\omega)| \cdot \omega$ product, provided that a top (flat) portion of the curve may be identified. Estimation of time domain parameters is even better for curves fitting underneath a trapezoid whose sides are steeper than ±20 dB/decade. The two cutoff frequencies (-3 dB points) of the top portion, if they can be identified, define the two poles α and β which are the main contributors of the duration and rise time of the pulse respectively. This method is NOT valid if the signal is narrowband, or becomes narrowband as a result of filtering or propagation through a transfer function.

It is still possible to get good results if a signal does not fully meet these guidelines, but results should be checked carefully.

3.3.3 Relating Power and Energy with the Frequency Domain

The method presented above is adequate when comparing against a threshold given in volts or amperes (which may be function of frequency), such as to determine the upset level of a component or system. However, device failure (burnout) is often the result of an overheating within a very small area, typically in the surrounding of the junction area of semiconductors. Not surprisingly, many theoretical and experimental studies (Refs. [20] [21] [22]) have shown a relation between the damage threshold of semiconductors and the power or energy of the pulse. Kalma (Ref. [20]) has shown that the failure level P_f can be related to the pulse width (T) as:

$$P_f = \frac{A}{T} + \frac{B}{\sqrt{T}} + C \tag{15}$$

where the term in T^{-1} corresponds to the adiabatic heating regime where the failure threshold is related to the total energy, the term in $T^{-½}$ corresponds to the quasi-adiabatic regime and the last term corresponds to the steady-state regime. Many authors (Refs. [21] [22]) consider only the second term as it often gives a good model for a wide range of pulse width, using:

$$P_f = \frac{A}{\sqrt{T}} \quad \text{or} \quad P_f = A \cdot T^{-B} \tag{16}$$

P_f may also be expressed in term of resistance (R) and voltage (V_f) or current (I_f). It has been shown in Section 3.3.2 that the $|H(\omega)|\cdot\omega$ curve can be used to obtain an estimation of the pulse width as $T = 0.7/\alpha$. By considering Equation (15) as three distinct regions, it may be written as[1]:

$$V_f = \left(1.2\sqrt{A}\sqrt{\alpha} + 1.1\sqrt{B}\sqrt[4]{\alpha} + \sqrt{C} \right)\cdot\sqrt{R} \tag{17}$$

which now relates the voltage (or current) threshold with the cutoff frequency α ($\alpha=2\pi f_c$). A signal will meet the threshold criteria V_f if its α point lies underneath this curve. Since this point also delimitates the left of the top portion of the $|H(\omega)|\cdot\omega$ curve and that V_f has zero or positive slope, the whole curve has to lie underneath the V_f threshold curve; therefore, the expression of the threshold V_f above may be plotted directly against frequency.

To illustrate this, experimental data from [22] is used to model the damage susceptibility of TTL logic family. Measurements were made with pulse width between 0.1 and 10 µs and the $P_f = AT^{-B}$ model was used. The steady-state term was estimated from data sheets. The resulting V_f threshold curve for both input and output device terminals is shown in Figure 13 (top). The measurements were not fast enough to estimate the adiabatic term; its approximate location is shown as a dotted line. The same curve, versus frequency, is shown on the bottom, along with three sample signals, just meeting the threshold criteria (the last one assuming an adiabatic term, in dotted line).

[1] The coefficients 1.1 and 1.2 could be ignored, introducing a small error of less than 0.8 dB for most frequencies, or 1.6 dB at the higher frequencies.

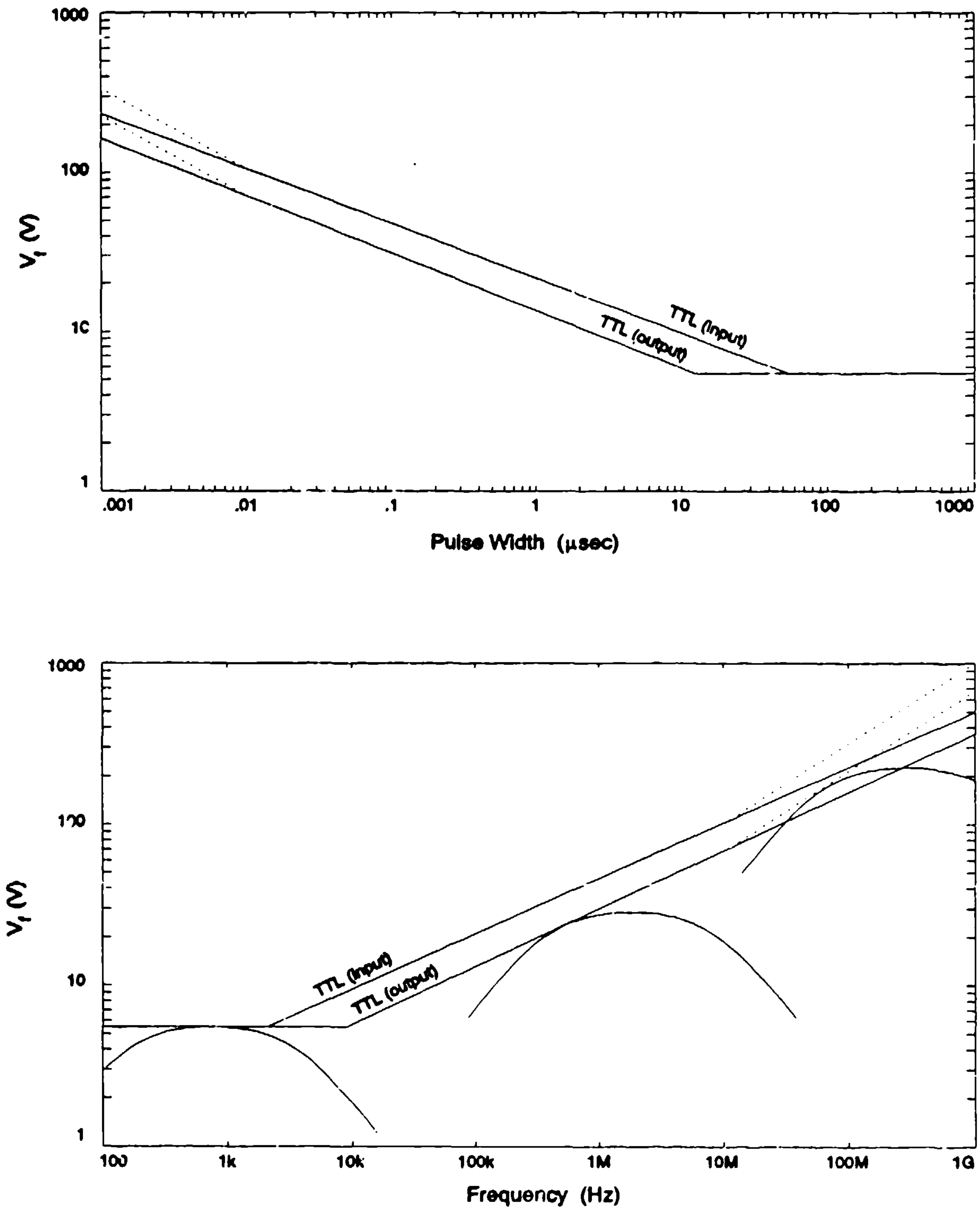

Figure 13. Failure voltage threshold of TTL logic family.

3.3.4 Propagation of Electromagnetic Attributes

As discussed above, fields and coupling interactions are categorized as either wideband or narrowband, yielding to four different types of interactions:

Field	Coupling	Result
wideband	wideband	wideband
wideband	narrowband	narrowband
narrowband	wideband	narrowband
narrowband	narrowband	narrowband

a) Wideband—wideband interactions: When both the field E(f) and the transfer function H(f) are wideband, the result R(f) is also wideband, and we simply add (or subtract)[1] the two functions together (when quantities are given in dB). When a graphical solution is desired, the product $|H(\omega)|\cdot\omega$ may be used and propagated through wideband transfer functions, allowing one to read the peak value and identify the frequency ranges of most susceptibilities at all steps.

b) Wideband—narrowband interactions: A wideband field E(f) propagated through a narrowband transfer function (of amplitude $H(f_r)$ and bandwidth bw_h) yields to a narrowband signal of amplitude $2\pi f_r E(f_r)+H(f_r)$ of bandwidth bw_h. It can be shown that this estimation of the peak value is accurate to ±3 dB in most cases, with a slightly higher error (± 6 dB) for cases where the spectrum of the field has a pronounced slope at f_r or when it is only marginally wideband (ie. the two predominant poles are close together).

c) Narrowband—wideband interactions: A narrowband field of amplitude $E(f_r)$ and bandwidth E_{bw} is propagated through a wideband transfer function H(f) by considering H(f) discrete at f_r, yielding a signal of amplitude $E(f_r)+H(f_r)$ and bandwidth E_{bw}. As in the case above, the error is quite small, but tend to grow as the slope of H(f) at f_r increases. However, the larger error at steeper slopes is usually not significant as the magnitude of H(f) in these areas is also much lower. This type of uncertainty can be easily accounted for by the use of confidence factors or fuzzy arithmetic described in Section 3.2.

d) Narrowband—narrowband interactions: Three different algorithms are required to model the propagation of a narrowband field $E(f_{re})$ of bandwidth bw_e through a narrowband transfer function $H(f_{rh})$ of bandwidth bw_h, depending on the

---

[1] The electromagnetic attributes are all stored in dB as defined in sections 3.1 and 3.3.1; therefore, all additions of field quantities in this section represent the multiplication of two transfer functions.

relation between f_{re} and f_{rh}: they may be in-band, off-band or out-of-band). For $f_{re}=f_{rh}$ (in-band), the result is a narrowband signal whose amplitude is given by:

$$(E(f_r)+H(f_r)) \cdot \min(bw_h/bw_e, 1) \tag{18}$$

and with a bandwidth $bw_r = \min(bw_e, bw_h)$. For $f_{re} \approx f_{rh}$ (off-band or marginally in-band) the bandwidth of the result, bw_r, is calculated as the overlap of the two functions E(f) and H(f), which is necessarily smaller than $\min(bw_e, bw_h)$, and the resonant frequency of the result, f_{rr}, is typically the resonant frequency of the narrowest of E(f) or H(f) or somewhat near the middle of the overlap if they are of comparable bandwidth. A very simple algorithm, assuming rectangular distribution of E(f) and H(f) around their respective resonant frequency of given bandwidth and resonant frequency, or a more sophisticated expressions, using trapezoidal or gaussian curves, can be used to calculate bw_r. The amplitude of the result is given by:

$$(E(f_r)+H(f_r)) \cdot bw_r/bw_e \tag{19}$$

For $f_{re} \neq f_{rh}$ (out-of-band), the result can still be significant. In order to estimate its value, it is necessary to know more about E(f) and H(f) outside their resonance region. The formulation used to describe narrowband signals could be refined, by using additional attributes, to specify the slope of the spectrum on both sides of f_r. For instance, the spectrum of a damped sinusoid has a slope of 0 on the left side of f_r and of -12 dB/octave on the right side. The result can be estimated as the largest of $E(f_{re})+H(f_{re})$ (of bandwidth bw_e) and $E(f_{rh})+H(f_{rh})$ (of bandwidth bw_e). In some cases, both expressions yield to results of comparable magnitude; where one could keep them both for further calculations.

As with the wideband interactions, it is also possible that E(f) and/or H(f) are specified over a range of frequencies. In that case, it is reasonable to assume the worst case, ie. that E(f) and H(f) are in-band and that the first algorithm as given in Equation (18) is used.

4.0 CONCLUSION

It has been have found in this study that HardSys/HardDraw may be a very valuable engineering tool to assist in the design of electromagnetic protection of electronics. Some deficiencies were identified and enhancements proposed to overcome them.

HardSys/HardDraw performs electromagnetic interactions analysis by using topological decomposition. HardDraw performs all the user interactions and includes a drawing tool to create or edit the electromagnetic topology. It is a major piece of work and a very good implementation of what a user-friendly interface should be. The primary negative comment about HardDraw is that it is written mostly in PostScript and based on a non-commercial platform, making it very difficult to modify or even to adapt to newer releases of the operating system. HardSys/HardDraw could also be greatly improved by adding some hardcopy capability.

HardSys is the advisor part of the expert system. It is knowledge-based, that is it contains a database of models and properties for various types of electromagnetic interactions. Problems are solved by using topology decomposition. HardSys takes into account the characteristics of electromagnetic emissions, the shielding effectiveness and the susceptibility of components to calculate the likelihood of failure of the system.

HardSys describes the electromagnetic quantities (ambient field, shielding effectiveness and system susceptibility) by using qualitative words associated with discrete levels. This formulation is inadequate for the calculation of electromagnetic attributes, unless a very large error margin can be tolerat d, resulting in a significant overdesign. It is also inadequate for the estimation of significant time domain parameters, such as peak value, rise time and duration.

An alternate definition of broadband and narrowband electromagnetic quantities was introduced, along with a set of algorithms (rules) to calculate their propagation. A new method for estimating significant time domain characteristics, such as peak value, rise time and duration, directly from the frequency domain was presented. This method can be applied for the calculation of failure index and shielding effectiveness. It is also suitable for the calculation of the optimal additional shielding required to protect components against upset or damage, based on threshold criteria defined in terms of voltage, current, power, energy or duration.

APPENDIX A

Prolog Implementation of Electromagnetic Attributes

```
:- prolog_flag(character_escapes,_,on).

%%%%
%   tf_evaluate( tf, tf_residuel, freq, tf_resultat )
%               Interpolate a tf at a given frequency
%
%   Arguments:  tf      transfert function,  see below
%               tf_resultat = (f/log(f),v;cf)
%               freq    frequency as f or (f/log(f))
%               tf_residuel
%                       used for optimisation
%
%   Notes:      Logarythmique interpolation is done, ie. F-axis is log. and
%               V-axis is linear in dB.
%
%   (tf) transfert functions:
%               List (ordered in f) whose elements describe a Bode plot.
%               Each element takes the form (f,v;cf) where f is the frequency,
%               v is the magnitude in dB and cf is a confidence factor [0..1].
%               cf is optional and is 1 by default.  f may also be given as
%               f/log(f) which speeds up the computation.
%
%               Function v or (v;cf) defines a constant.
%%%%

tf_log10(0,-100) :- !.                          % Take care of f=0 (DC)
tf_log10(X,Y) :- log10(X,Y).

tf_evaluate( TF, TFR, FRQ, (F/FL,VI;CFI) ) :-
    nonvar(TF),

    TF = [E1|TFT],  TFT=[E2|_],                 % Expand 2 first elements
    ( E1=(FRQ1,V1;CF1) ; E1=(FRQ1,V1),CF1=1 ),
    ( E2=(FRQ2,V2;CF2) ; E2=(FRQ2,V2),CF2=1 ),
    ( FRQ1=F1/FL1 -> true ; F1=FRQ1 ),
    ( FRQ2=F2/FL2 -> true ; F2=FRQ2 ),

    ( FRQ=(F/FL) -> true ; F=FRQ, tf_log10(F,FL) ),

    ( F1 ~= F2 ->
        format('~N! Error in tf_evaluate\c
                ~n! Skipping duplicate frequency ~f',F1),
        tf_evaluate( TFT,TFR,(F/FL),(F/FL,VI;CFI))

    ; F1 > F2 ->
        format('~N! Fatal error in tf_evaluate\c
                ~n! Frequency list out of order~n! Execution aborted',_),
        fail

    ; F ~= F1 ->                                % F=F1
        VI= V1, CFI= CF1

    ; F ~= F2 ->                                % F=F2
        VI= V2, CFI= CF2
```

```
    ; F < F1 ->                                    % Extrapolation at beginning
        ( nonvar(FL1) ; tf_log10(F1,FL1), tf_log10(F2,FL2) ),
        Ft is (FL-FL1)/(FL2-FL1),
        VI  is (V2-V1)*Ft + V1,                    % Extrapolate segment @F1-F2
        CFI is CF1 * F/F1                          % Compute CF

    ; F > F2, TFT=[_] ->                           % Extrapolation at end
        ( nonvar(FL1) ; tf_log10(F1,FL1), tf_log10(F2,FL2) ),
        Ft is (FL-FL1)/(FL2-FL1),
        VI  is (V2-V1)*Ft + V1,                    % Extrapolate segment @F1-F2
        CFI is CF2 * F2/F                          % Compute CF

    ; F < F2 ->                                    % Interpolate segment @F1-F2
        ( nonvar(FL1) ; tf_log10(F1,FL1), tf_log10(F2,FL2) ),
        Ft is (FL-FL1)/(FL2-FL1),
        VI  is (V2-V1)*Ft + V1,
        CFI is (CF2-CF1)*Ft + CF1

    ; % else
        tf_evaluate(TFT,TFR,(F/FL),(F/FL,VI;CFI))
    ),

    ( nonvar(TFR) ; TF=TFR ), !.


tf_evaluate( TF, TF, FRQ, (F/FL,VI;CFI) ) :-    % TF=constant
    nonvar(TF),
    ( TF=(VI;CFI) -> true ; VI=TF, CFI=1 ),
    ( FRQ=(F/FL)  -> true ; F=FRQ, tf_log10(F,FL) ), !.

tf_evaluate(TF,_,_,_) :-
    var(TF), !,
    format('~N! Fatal error in tf_evaluate\c
          ~n! 1st argument must be instantiated',_),
    fail.

tf_evaluate(_,_,_,_) :-
    format('~N! fatal unknown error in tf_evaluate',_).


%%%%
%   tf_logf( TF1, TF2 )
%               Compute log10 of frequency if necessary
%
%   Arguments:  TF1     as [ (F,V;CF), ...]      (CF optional)
%               TF2     as [ (F/log(F),V;CF), ...]
%%%%

tf_logf( [E1|T1], [(F1/FL1,V1;CF1)|T2] ) :-
    ( E1=(FRQ1,V1;CF1) ; E1=(FRQ1,V1),CF1=1 ),
    ( FRQ1=F1/FL1 -> true ; F1=FRQ1, tf_log10(F1,FL1) ),
    tf_logf(T1,T2).

tf_logf( K, (V,CF) ) :-
    K=(V,CF) ; number(K), V=K, CF=1.

tf_logf( [], [] ).
```

```
%%%%
%   tf_xeq( (v1;cf1), (v2;cf2), (v3;cf3), xeq )
%               Internal routine
%
%   Arguments:  xeq=(xeq_v,xeq_cf)  functions to compute v3 and cf3
%                  =(cf,op)         use cf/4 (see cf.pl)
%                  =xeq             single function working with (v;cf) notation
%%%%

tf_xeq( (V1;CF1), (V2;CF2), (V3;CF3), XEQ ) :-
    atom(XEQ) ->                            % fct working with (v;cf)
        XQ =.. [XEQ,(V1;CF1),(V2;CF2),(V3;CF3)],
        call(XQ) ;

    XEQ=(cf,OP) ->                          % Use cf/4
        cf(OP,(V1;CF1),(V2;CF2),(V3;CF3))

    ; /* else */                            % 2 fcts for v and cf
        XEQ=(XEQV,XEQCF),
        XQV =.. [XEQV,V1,V2,V3],      call(XQV),
        XQC =.. [XEQCF,CF1,CF2,CF3], call(XQC).


%%%%
%   tf_propagate( tf1, tf2, tf3, (xeq_v,xeq_cf) )
%
%   Arguments:  Result tf3 is function of tf1 and tf2
%               xeq_v & xeq_cf, calls to evaluate V and CF (see tf_xeq)
%
%   Notes:      tf3 is computed for every frequencies of tf1 and tf2
%               tf1 and/or tf2 may be constants (CF must be specified)
%%%%

% tf_xpropagate/4  --  Initialisation

tf_xpropagate( TF1, TF2, (V3;CF3), XEQ ) :-            % tf1 & tf2 constants
    ( TF1=(V1;CF1) ; number(TF1), V1=TF1, CF1=1 ),
    ( TF2=(V2;CF2) ; number(TF2), V2=TF2, CF2=1 ), !,
    tf_xeq((V1;CF1),(V2;CF2),(V3;CF3),XEQ).

tf_xpropagate( TF1, TF2,  TF3, XEQ ) :-                % tf2 constant
    ( TF2=(V2;CF2) ; number(TF2), V2=TF2, CF2=1 ), !,
    tf_logf(TF1,TFL1),
    tf_xpropagate( TFL1, TFL1, [], (V2;CF2), TF3, XEQ ).

tf_xpropagate( TF1, TF2, TF3, XEQ ) :-                 % tf1 constant
    ( TF1=(V1;CF1) ; number(TF1), V1=TF1, CF1=1 ), !,
    tf_logf(TF2,TFL2),
    tf_xpropagate( [], (V1,CF1), TFL2, TFL2, TF3, XEQ ).

tf_xpropagate(  TF1, TF2, TF3, XEQ ) :-                % tf1 & tf2 standard
    tf_logf(TF1,TFL1), tf_logf(TF2,TFL2),
    tf_xpropagate( TFL1, TFL1, TFL2, TFL2, TF3, XEQ ).


% tf_xpropagate/6  --  Serious work begins here...

tf_xpropagate( [], _, [], _, [], _ ) :- !.       % Fin

tf_xpropagate( [(F1/FL1,V1;CF1)|TF1T], TF1R, [(F2/_,V2;CF2)|TF2T], TF2R,
               TF3, XEQ ) :-
    F1 =:= F2, !,
    tf_xeq((V1;CF1),(V2;CF2),(V5;CF5),XEQ),
    TF3 = [ (F1/FL1,V5;CF5) | TF3T ],
    tf_xpropagate( TF1T, TF1R, TF2T, TF2R, TF3T, XEQ ).
```

```
tf_xpropagate( [(F1/FL1,V1;CF1)|TF1T], TF1R, TF2, TF2R,
                TF3, XEQ ) :-
    ( TF2 = [] ;
      TF2 = [(F2/_,_;_)|_], nonvar(F1), nonvar(F2), F1 < F2 ), !,

    tf_evaluate(TF2R,TF2R2,(F1/FL1),(F1/FL1,V2;CF2)),
    tf_xeq((V1;CF1),(V2;CF2),(V5;CF5),XEQ),
    TF3 = [ (F1/FL1,V5;CF5) | TF3T ],
    tf_xpropagate( TF1T, TF1R, TF2, TF2R2, TF3T, XEQ ).

tf_xpropagate( TF1, TF1R, [(F2/FL2,V2;CF2)|TF2T], TF2R,
                TF3, XEQ ) :-
    ( TF1 = [] ;
      TF1 = [(F1/_,_;_)|_], nonvar(F1), nonvar(F2), F1 > F2 ), !,
    tf_evaluate(TF1R,TF1R2,(F2/FL2),(F2/FL2,V1;CF1)),
    tf_xeq((V1;CF1),(V2;CF2),(V5;CF5),XEQ),
    TF3 = [ (F2/FL2,V5;CF5) | TF3T ],
    tf_xpropagate( TF1, TF1R2, TF2T, TF2R, TF3T, XEQ ).
```

APPENDIX B

Prolog Implementation of Confidence Factors Arithmetic

```
%%%%
%   Equivalence:
%              CF=       1   .9   .8   .7   .6   .5   .4   .3   .2   .1   .0
%              Err(dB)=  0   6    12   18   24   30   36   42   48   54   60-
%
%                                                     3 CF
%              CF = 1-CF     E       = 60 CF     E = 10          E     = 100(E -1)
%                             r(dB)                r               r(%)          r
%%%%

:- current_op(P,xfx,is), op(P,xfx,is_cf).        % R is expression
:- current_op(P,yfx,+),  op(P,yfx,+@),           % add, in dB
                         op(P,yfx,+//@).         % parallel add, in dB
:- current_op(P,yfx,*),  op(P,yfx,*@).           % multiply, in dB
:- current_op(P,yfx,/),  op(P,yfx,/@)            % divide, in dB

is_cf((V;CF),(V;CF)) :- !.
is_cf((V;1),V) :- number(V), !.

is_cf(R,CTerm) :-                                % Expression parser
    CTerm =.. [Op,A1,A2],
    A1R is_cf A1,  A2R is_cf A2,
    CTerm2=.. [cf,Op,A1R,A2R,R],
    (call(CTerm2)->true).

cf(+, (V1;CF1), (V2;CF2), (Vr;CFr) ) :-          % Add
    Vr is V1+V2,
    ( V1>V2 ->
        ( V1>10*V2 ->   CFr=CF1
                   ;    K is 0.5 * V2/V1,
                        CFr is max(K*(CF2-CF1)+CF1,0) )
    :   ( V2>10*V1 ->   CFr=CF2
                   ;    K is 0.5 * V1/V2,
                        CFr is max(K*(CF1-CF2)+CF2,0) )
    ).

cf(+@, (V1;CF1), (V2;CF2), (Vr;CFr) ) :-         % Add, in dB
    dB_add(V1,V2,Vr),
    ( V1>V2 ->
        ( V1-V2>20 ->   CFr=CF1
                   ;    K is 0.5 - 0.0225*(V1-V2),
                        CFr is max(K*(CF2-CF1)+CF1,0) )
    :   ( V2-V1>20 ->   CFr=CF2
                   ;    K is 0.5 - 0.0225*(V2-V1),
                        CFr is max(K*(CF1-CF2)+CF2,0) )
    ).

cf(+//@, (V1;CF1), (V2;CF2), (Vr;CFr) ) :-       % Parallel add, in dB
    cf(+@, (-V1;CF1), (-V2;CF2), (Vt;CFr) ),
    Vr is -Vt.

cf(*, (V1;CF1), (V2;CF2), (Vr;CFr) ) :-          % Multiply
    Vr  is V1*V2,
    CFr is max(CF1+CF2-1,0).
```

```
cf(*@, (V1;CF1), (V2;CF2), (Vr;CFr) ) :-          % Multiply, in dB
    Vr  is V1+V2, % dB_mul(V1,V2,Vr)
    CFr is max(CF1+CF2-1,0).

cf(/, (V1;CF1), (V2;CF2), (Vr;CFr) ) :-           % Divide
    Vr  is V1/V2,
    CFr is max(CF1+CF2-1,0).

cf(/@, (V1;CF1), (V2;CF2), (Vr;CFr) ) :-          % Divide, in dB
    Vr  is V1-V2, % dB_div(V1,V2,Vr)
    CFr is max(CF1+CF2-1,0).
```

APPENDIX C

Prolog Implementation of Fuzzy Logic

```
%%%%
%   Reference:  Introduction to fuzzy arithmetic, Kaufmann & Gupta
%
%   Notes:      Trapezoidal definition is used, as (a1,a2,a3,a4)
%
%               Definition of some operators is not exact (see Kaufmann).
%
%   Syntax:     Use in-fix notation (eg. R -: 10 * (3,4,5,6) )
%%%%

:- current_op(P,xfx,is), op(P,xfx,-:).             % assign
:- current_op(P,_,mod),   op(P,xfy,**).            % power of

-:((A1,A2,A3,A4),(A1,A2,A3,A4)) :-  !.             % fuzzy number: trapezoidal
-:((A,A,A,A),A) :-  number(A), !.                  % scalar, convert to fuzzy

-:(R,CTerm) :-                                     % Monadic operators
    CTerm =.. [Op,A1],
    A1R -: A1,
    CTerm2=.. [fuzzy,Op,A1R,R],
    (call(CTerm2)->true).                          % Cut - one answer only

-:(R,CTerm) :-                                     % Diadic operators
    CTerm =.. [Op,A1,A2],
    A1R -: A1, A2R -: A2,
    CTerm2=.. [fuzzy,Op,A1R,A2R,R],
    (call(CTerm2)->true).                          % Cut - one answer only


%%%%
%   fuzzy/3     Call for monadic operators
%
%   Notes:      Operators such as sqrt, exp, log & log10 are implemented with
%               the catch all entry.
%%%%

fuzzy(-,(A1,A2,A3,A4),(R1,R2,R3,R4)) :-            % Negation
    R1 is -A4,  R2 is -A3,  R3 is -A2,  R4 is -A1.

fuzzy(OP,A,R) :-                                   % catch-all
    A= (A1,A2,A3,A4),  AL= [A1,A2,A3,A4],
    R= (R1,R2,R3,R4),  RL= [R1,R2,R3,R4],
    math_list(OP,AL,RL).

%%%%
%   fuzzy/4     Call for diadic operators
%%%%

fuzzy(+, (A1,A2,A3,A4), (B1,B2,B3,B4),             % Addition
       (R1,R2,R3,R4)) :-
    R1 is A1+B1,  R2 is A2+B2,  R3 is A3+B3,  R4 is A4+B4.

fuzzy(-, (A1,A2,A3,A4), (B1,B2,B3,B4),             % Subtraction
       (R1,R2,R3,R4)) :-
    R1 is A1-B4,  R2 is A2-B3,  R3 is A3-B2,  R4 is A4-B1.
```

```
fuzzy(*, (A1,A2,A3,A4), (B1,B2,B3,B4),          % Multiplication
    (R1,R2,R3,R4)) :-
    A1 >= 0, B1 >= 0,
    R1 is A1*B1,  R2 is A2*B2,  R3 is A3*B3,  R4 is A4*B4.

fuzzy(/, (A1,A2,A3,A4), (B1,B2,B3,B4),          % Division
    (R1,R2,R3,R4)) :-
    A1 >= 0, B1 >= 0,
    R1 is A1/B4,  R2 is A2/B3,  R3 is A3/B2,  R4 is A4/B1.

fuzzy(**, (A1,A2,A3,A4), (B1,B2,B3,B4), (R1,R2,R3,R4)) :- % power of
    pow(A1,B1,R1), pow(A2,B2,R2), pow(A3,B3,R3), pow(A4,B4,R4).

fuzzy(min, (A1,A2,A3,A4), (B1,B2,B3,B4), (R1,R2,R3,R4)) :- % minimum
    min(A1,B1,R1), min(A2,B2,R2), min(A3,B3,R3), min(A4,B4,R4).

fuzzy(max, (A1,A2,A3,A4), (B1,B2,B3,B4), (R1,R2,R3,R4)) :- % maximum
    max(A1,B1,R1), max(A2,B2,R2), max(A3,B3,R3), max(A4,B4,R4).

fuzzy(OP,A,B,R) :-                              % catch-all
    A= (A1,A2,A3,A4),  AL= [A1,A2,A3,A4],
    B= (B1,B2,B3,B4),  BL= [B1,B2,B3,B4],
    R= (R1,R2,R3,R4),  RL= [R1,R2,R3,R4],
    math_list(OP,AL,BL,RL).

%%%%
%   fuzzy(error,        Add an error (en per-cent) to a fuzzy number
%%%%

fuzzy(error, (A1,A2,A3,A4), (E,E,E,E),
    (R1,R2,R3,R4)) :-
    E200 is E/200, E100 is 2*E200,              % E/100 & E/200
    R1 is A1*(1-E100),  R2 is A2*(1-E200),      % trapez. (-10% -5% +5% +10%)
    R3 is A3*(1+E200),  R4 is A4*(1+E100).
```

REFERENCES

[1] J. LoVetri, "*HardSys/HardDraw Electromagnetic Interaction Modelling*", Final report for Development Project D6002-02171, DASP, May 1991

[2] J. LoVetri, G.I. Costache, "*An Electromagnetic Interaction Modeling Advisor*", IEEE Transaction on Electromagnetic Compatibility, Vol. 33, Nº 3, Aug. 1991

[3] J. LoVetri, D.P.W. Graham, "*Constraint Propagation throaugh Electromagnetic Interaction Topologies*", ANTEM'90, Symposium on Antenna Tech. and Appl. Electromagnetics

[4] J. LoVetri, S. Abu-Hakima, A.S. Podgorski, G.I. Costache, "*HardSys: Applying Expert System Techniques to Electromagnetic Hardening*", IEEE 1989 National Symposium on Electromagnetic Compatibility

[5] S.B. Kass, "*Proposed Enhancements to the HardSys Electromagnetic Hardening Advisor*", National Research Concil Canada, March 1991

[6] J. LoVetri, W.H. Henneker, "*Fuzzy Logic Implementation of an Electromagnetic Interactions Modeling Tool*", National Research Concil Canada

[7] C.E. Baum, "*Electromagnetic Topology for the Analysis and Design of Complex Electromagnetic Systems*", Proceedings of the NATO Advanced Study Institute on fast Electrical and Optical Diagnostic Principles and Techniques, 1986

[8] C.E. Baum, "*Electromagnetic Topology: A formal Approach to the Analysis and Design of Systems*", Interaction Notes, Note 400, Air Force Weapons Lab, 1980

[9] M.A. Messier, "*EMP Hardening Topology Expert System (Hard Top)*", Interaction Notes, Note 449, Air Force Weapons Lab, June 1985

[10] F.M. Tesche, "*Topological Concepts for Internal EMP Interaction*", IEEE Transaction on Electromagnetic Compatibility, Vol. 20, Nº 1, Aug. 1978

[11] F.M. Tesche, "*Introduction to Concepts of Electromagnetic Topology as Applied to EMP Interaction with Systems*", AGARD Lecture Series Nº 144, 1986

[12] Quintus Prolog Development System Manuals, Release 3.1.1 for Sun-3 & Sun-4, 1991

[13] E.P. Stabler, "*Object-Oriented Programming in Prolog*", AI Expert, Oct 1986

[14] D.R.J. White, M. Mardiguian, "*EMI Control - Methodology and Procedures*", 4th edition, Interference Control Technologies, 1985

[15] D.R.J. White, "*Shielding Design - Methodology and Procedures*", 4th edition, Interference Control Technologies, 1985

[16] S. Haykin, "*Communication Systems*", John Wiley & Sons, 1978

[17] K.S.H. Lee, "*EMP Interaction: Principles, Techniques and Reference Data*", Hemisphere Publishing Corporation, 1986

[18] "*Electromagnetic Emission and Susceptibility Requirements for the Control of Electromagnetic Interference*, Military standard 461C, deparment of Defence, USA, 1986

[19] A. Kaufmann, M.M. Gupta, "*Introduction to Fuzzy Arithmetic*", Van Nostrand Reinhold Company, 1985

[20] A.H. Kalma, C.J. Fischer, "*Electrical Pulse Burnout Testing of Light-Emitting Diodes*", IEEE Transactions on Nuclear Science, Vol. NS-22, Nº 6, Dec 1975

[21] D.C. Wunsch, R.R. Bell, "*Determination of Threshold Failure Levels of Semiconductor Diodes and Transistors Due to Pulse Voltages*", IEEE Transactions on Nuclear Science, Vol. NS-15, Nº 6, Dec 1968

[22] C.R. Jenkins, D.L. Durgin, "*EMP Susceptibility of Integrated Circuits*", IEEE Transactions on Nuclear Science, Vol. NS-22, Nº 6, Dec 1975

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. **ORIGINATOR** (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.) Defence Research Establishment Ottawa Ottawa, Ontario K1A 0Z4	2. **SECURITY CLASSIFICATION** (overall security classification of the document including special warning terms if applicable) **UNCLASSIFIED**

3. **TITLE** (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

Evaluation of HardSys/HardDraw, an Expert System for Electromagnetic Interactions Modelling (U)

4. **AUTHORS** (Last name, first name, middle initial)

Dion, M.

5. **DATE OF PUBLICATION** (month and year of publication of document) May 1993	6a. **NO. OF PAGES** (total containing information. Include Annexes, Appendices, etc.) 56	6b. **NO. OF REFS** (total cited in document) 22

7. **DESCRIPTIVE NOTES** (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.

DREO Report

8. **SPONSORING ACTIVITY** (the name of the department project office or laboratory sponsoring the research and development. Include the address.

DMEE-7

9a. **PROJECT OR GRANT NO.** (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant) Project 041LT	9b. **CONTRACT NO.** (if appropriate, the applicable number under which the document was written)
10a. **ORIGINATOR'S DOCUMENT NUMBER** (the official document number by which the document is identified by the originating activity. This number must be unique to this document.) DREO REPORT 1175	10b. **OTHER DOCUMENT NOS.** (Any other numbers which may be assigned this document either by the originator or by the sponsor)

11. **DOCUMENT AVAILABILITY** (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. **DOCUMENT ANNOUNCEMENT** (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). however, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited Announcement

13. **ABSTRACT** (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

The department of National Defence and the National Research Council have sponsored the development of HardSys/HardDraw, an expert system for the modelling of electromagnetic interactions in complex systems. This report gives a description of HardSys/HardDraw and reviews the main concepts used in its design. Various aspects of its implementation, user interaction and modelling concepts are evaluated. Some deficiencies are identified and enhancements are proposed to overcome them. Concepts of uncertainty are reviewed and an approach using confidence factors and fuzzy arithmetic is developed. A new method relating both the frequency and time domains is presented and is applied for the calculation of failure index and shielding effectiveness.

Le département de la défense nationale et le centre national de recherche ont développé un système expert pour prédire les interactions électromagnétiques dans des systèmes complexes. Ce rapport donne une description de ce système et des concepts utilisés. Plusieurs aspect de son implémentation, de son interaction avec l'utilisateur et des modèles utilisés sont discutés. Certaines déficiences sont identifiées et plusieurs améliorations sont proposées. Le concept d'incertitude est présenté et une approche utilisant les facteurs de confiance et l'arithmétique floue est présentée. Une approche innovatrice pour relier les domaines fréquentiel et temporel est présentée et est appliquée pour le calcul des indices de défaillance ou des coefficients de blindage électromagnétique.

14. **KEYWORDS, DESCRIPTORS or IDENTIFIERS** (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Confidence Factors
Damage & Upset Thresholds
Electromagnetic Interactions
Electromagnetic Propagation
Expert System
Fuzzy Arithmetic
HardDraw
HardSys
Shielding Effectiveness
(Electromagnetic) System Susceptibility
(Electromagnetic) Topological Decomposition